

# ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra radioelektroniky

## DIPLOMOVÁ PRÁCE

Aplikace komprimovaného snímání ve zpracování  
obrazu

Applications of compressive sensing in image  
processing

Studijní program: Komunikace, multimedia a elektronika

Studijní obor: Multimediální technika

Vedoucí práce: Ing. Karel Fliegel, Ph.D.

2018

Filip Andrásy

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Andrássy** Jméno: **Filip** Osobní číslo: **411195**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra radioelektroniky**  
Studijní program: **Komunikace, multimédia a elektronika**  
Studijní obor: **Multimediální technika**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Aplikace komprimovaného snímání ve zpracování obrazu**

Název diplomové práce anglicky:

**Applications of compressive sensing in image processing**

Pokyny pro vypracování:

Podejte přehled metod komprimovaného snímání a jejich aplikací ve zpracování obrazu. Zaměřte se zejména na metody pro kompresi a rekonstrukci obrazu. Vybrané algoritmy implementujte a ověřte jejich účinnost na základě počítačových simulací.

Seznam doporučené literatury:

- [1] Elad, M.: Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing, Springer, 2010.  
[2] Ahn, J. H.: Compressive Sensing and Recovery for Binary Images, IEEE Transactions on Image Processing, vol. 25, no. 10, 2016.

Jméno a pracoviště vedoucí(ho) diplomové práce:

**Ing. Karel Fliegel, Ph.D., katedra radioelektroniky FEL**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **18.09.2017**

Termín odevzdání diplomové práce: \_\_\_\_\_

Platnost zadání diplomové práce:

**do konce zimního semestru 2018/2019**

\_\_\_\_\_  
Podpis vedoucí(ho) práce

\_\_\_\_\_  
Podpis vedoucí(ho) ústavu/katedry

\_\_\_\_\_  
Podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta

## **Čestné prohlášení**

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 09.01.2018

.....  
Filip Andrassy

## **Pod'akovanie**

Rád by som sa poďakoval pánovi Ing. Karlovi Fliegelovi, Ph.D. za jeho odborné rady a cenné pripomienky, ktorými prispel k vypracovaniu tejto diplomovej práce.

## Abstrakt

Táto diplomová práca sa zaoberá tematikou komprimovaného snímania. Cieľom práce je podať prehľad metód komprimovaného snímania a ich aplikáciu v spracovaní obrazu. V teoretickej časti je podaný úvod do problematiky riedkej reprezentácie signálov, kde sú zhrnuté základné definície, ktoré sú nevyhnutné pre určenie vhodného a jedinečného  $k$ -riedkeho riešenia systému nedourčených lineárnych rovníc. Na nájdenie riedkeho riešenia existuje množstvo metód. Niekoľko základných metód je uvedených a stručne popísaných v tejto práci. V praktickej časti bolo za úlohu navrhnuť a implementovať simuláciu jednopixelovej kamery. Motiváciou pre výber práve tejto aplikácie komprimovaného snímania bol fakt, že táto kamera má síce len jeden fotónový detektor, ale dokáže zrekonštruovať aj obraz s vyšším rozlíšením. Na rekonštrukciu obrazu je použitá metóda Orthogonal Matching Pursuit (OMP). Tento algoritmus bol vybraný pre jeho dobrý pomer medzi výpočtovou náročnosťou a výslednou kvalitou zrekonštruovaného obrazu.

## Kľúčové slová

Komprimované snímanie, riedka reprezentácia signálu,  $k$ -riedke riešenie, systém nedourčených lineárnych rovníc, jednopixelová kamera, simulácia, rekonštrukcia obrazu, MATLAB

## **Abstract**

This diploma thesis deals with the topic of compressive sensing. The aim of this work is to give an overview of compressive sensing methods and their application in image processing. Theoretical part presents an introduction to the problem of sparse representation of signals, where are summarized the basic definitions, which are necessary for determination of a suitable and unique  $k$ -sparse solution of the underdetermined linear system. There are many methods how to find a sparse solution. Several of the basic methods are mentioned and briefly described in this work. In the practical part, it was the task to design and implement the simulation of single-pixel camera. The motivation for choosing this application of compressive sensing was the fact, that this camera has only single photon detector, but is able to reconstruct an image with higher resolution. As the image reconstruction algorithm, there is used Orthogonal Matching Pursuit (OMP). This algorithm was chosen for its good ratio between the computational costs and the quality of the reconstructed image.

## **Key words**

Compressive sensing, sparse signal representation,  $k$ -sparse solution, underdetermined linear system, single-pixel camera, simulation, image reconstruction, MATLAB

# Obsah

Úvod .....	9
1 Riedka reprezentácia signálov .....	11
1.1 Definícia úlohy .....	11
1.2 Podmienky pre jednoznačnosť riešenia.....	12
1.2.1 Spark matice .....	12
1.2.2 Vzájomná koherencia (mutual coherence).....	12
1.2.3 Slovník vytvorený zlúčením dvoch ortobáz (špeciálny prípad).....	13
1.3 Výpočtová zložitosť problému .....	13
1.4 Podmienky ekvivalencie riešenia $\mathbf{l}_0$ - a $\mathbf{l}_1$ -minimalizáciou.....	13
1.4.1 NSP - vlastnosť nulového priestoru (Null Space Property).....	13
1.4.2 RIP - vlastnosť oslabenej izometrie (Restricted Isometry Property).....	14
2 Komprimované snímanie.....	15
3 Rekonštrukčné metódy komprimovaného snímania .....	17
3.1 Relaxačné algoritmy.....	18
3.1.1 Rekonštrukcia technikami $\mathbf{l}_1$ -minimalizácie.....	18
3.1.2 Basis Pursuit (BP) .....	20
3.1.3 Iterative Reweighted Least Squares (IRLS) .....	20
3.2 Hladné algoritmy (Greedy pursuits) .....	20
3.2.1 Všeobecný postup algoritmov typu MP/OMP .....	21
3.2.2 Matching Pursuit (MP).....	22
3.2.3 Orthogonal Matching Pursuit (OMP).....	23
3.2.4 Directional/Gradient Pursuit (GP) .....	24
3.2.5 Conjugate Gradient Pursuit (CGP) .....	24
3.2.6 Stagewise Orthogonal Matching Pursuit (StOMP) .....	25
3.2.7 Iterative Hard Thresholding (IHT) .....	26
3.3 Hybridné algoritmy.....	27
3.3.1 Compressive Sampling Matching Pursuit (CoSaMP) a Subspace Pursuit (SP) .....	27
3.3.2 A* Orthogonal Matching Pursuit (A*OMP) .....	27
4 Aplikácie komprimovaného snímania.....	28
4.1 Potlačovanie šumu (denoising).....	28
4.2 Doostrovanie (deblurring) .....	28
4.3 Jednopixelová CS kamera .....	29
4.3.1 Výhody jednopixelovej kamery .....	31

4.3.2 Nevýhody jednopixelovej kamery .....	31
4.3.3 Matematický popis jednopixelovej kamery.....	32
5 Praktická časť .....	33
5.1 Popis modelu .....	33
6 Vyhodnotenie výsledkov.....	35
6.1 Algoritmus OMP .....	35
6.2 Simulácie jednopixelovej kamery .....	39
6.3 Rekonštrukcia HD obrazu.....	45
6.4 Porovnanie jednopixelovej kamery so štandardnou kamerou.....	47
Záver .....	49
Zdroje.....	51
Zoznam použitých skratiek .....	53
Zoznam obrázkov.....	54
Zoznam tabuliek .....	55
Zoznam príloh.....	56



# Úvod

V súčasnej dobe je trendom vyrábať kamery, ktoré majú stále väčšie rozlíšenie. Štandardná kamera obsahuje obrovské pole fotónových detektorov, každý jeden detektor odpovedá jednému pixelu v obraze. Aktuálne sa pohybujeme pri rozlíšení v jednotkách až desiatkach miliónov pixelov.

Tradičný prístup pri rekonštrukcii signálov využíva všeobecne známy Nyquistov vzorkovací teorém. Ten nám hovorí, že presná rekonštrukcia signálu je možná za predpokladu, že je dodržaná podmienka  $f_{vz} > 2 f_{max}$ , kde  $f_{vz}$  je vzorkovacia frekvencia a  $f_{max}$  je frekvencia maximálna. Tento princíp sa využíva vo väčšine súčasných zariadení. V niektorých prípadoch, napr. v obrazovej technike, môže byť vzorkovacia frekvencia zbytočne vysoká, čo vedie k omnoho väčšiemu počtu vzoriek ako je potrebné. Dôsledkom toho sa pracuje s oveľa väčšími dátovými tokmi [3].

Teória komprimovaného snímania hovorí, že pokiaľ je signál riedky a komprimovateľný, tak môže byť rekonštruovaný z menšieho počtu vzoriek signálu. Komprimované snímanie využíva úplne iný prístup, v ktorom sa signál sníma lineárne a neadaptívne (viz kap. 2). To má za dôsledok, že signál snímame iba toľko krát, koľko je potrebné k jeho rekonštrukcii. Komprimované snímanie je jednou z dôležitých aplikácií v oblasti riedkych reprezentácií signálov [3].

Komprimované snímanie si postupne nachádza uplatnenie pri snímaní a spracovaní obrazu. Napríklad sa využíva pri jednopixelovej kamere, potlačovaní šumu (denoising), doostrovaní (deblurring), magnetickej rezonancii (MRI), dopĺňovaní chýbajúcej informácie v signály (inpainting) a podobne [13].

Jednopixelová kamera má na rozdiel od štandardnej kamery, len jeden fotónový detektor. Táto koncepcia patrí do širokej kategórie multiplexných snímacích metód, v ktorých sa vykonáva séria postupných meraní jediným detektorom. To čo však vyzdvihuje jednopixelovú CS kameru, je práve jej prístup vo vzorkovaní na základe teórie komprimovaného snímania. Z teórie vyplýva, že na úspešnú rekonštrukciu signálu postačí omnoho menší počet nasnímaných vzoriek ako je ich celkový počet [14]. Navrhnuť a implementovať simuláciu práve takéhoto modelu jednopixelovej kamery bolo cieľom praktickej časti.

Členenie dokumentu je nasledovné. Na začiatku práce je uvedený popis problematiky riedkej reprezentácie signálov z teoretického hľadiska, kde je zhrnutých pár základných definícií, ktoré sú nevyhnutné pre určenie vhodného a jedinečného  $k$ -riedkeho riešenia systému nedourčených lineárnych rovníc. V druhej kapitole je popísaný základný princíp komprimovaného snímania a jeho následná rekonštrukcia technikami  $l_1$ -minimalizácie. V tretej kapitole je rozdelenie algoritmov do základných skupín, pričom pri jednotlivých algoritmoch je aj stručne uvedený ich základný princíp. Štvrtá kapitola sa zaoberá aplikáciou komprimovaného snímania pri snímaní a spracovaní obrazu. Praktická časť práce začína od

piatej kapitoly, ktorá obsahuje popis modelu simulácie jednopixelovej kamery a popis základných parametrov simulácie. V šiestej kapitole sú znázornené dosiahnuté výsledky spolu s ukázkami zrekonštruovaných obrázkov. Na konci práce je záver, ktorý obsahuje zhrnutie celej tejto práce.

# 1 Riedka reprezentácia signálov

Túto problematiku z pohľadu matematiky je možné chápať ako systém nedourčených lineárnych rovníc, teda rovníc, ktorých počet je menší ako počet neznámych. V množine riešení týchto rovníc hľadáme riešenie, ktoré má čo najviac nulových hodnôt, teda tzv. riedke riešenie [4].

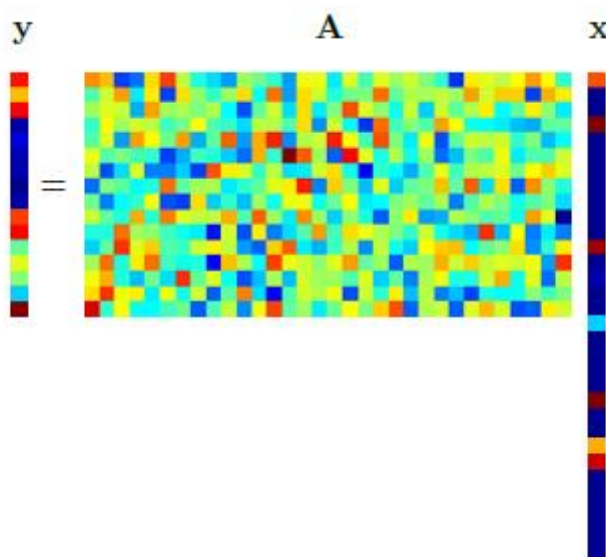
## 1.1 Definícia úlohy

Základnú úlohu popíšeme ako sústavu lineárnych rovníc v tvare  $\mathbf{Ax} = \mathbf{y}$ , kde matica  $\mathbf{A}$  je typu  $m \times N$ , kde  $m < N$ . Túto maticu nazývame slovník (dictionary) a jej stĺpce atómy (atoms), pričom občas sa nazýva aj matica meraní (measurement matrix). Vektor  $\mathbf{x}$  je neznámy. Vektor  $\mathbf{y}$  tvoria vstupné dáta [4,6]. Ilustratívna schéma je na obr. 1.

Neznámy, hľadaný vektor  $\mathbf{x}$  má byť čo najredší, takže by mal obsahovať čo najviac nulových zložiek. To môžeme zapísať ako

$$\min_x \|\mathbf{x}\|_0 \text{ vzhľadom k } \mathbf{Ax} = \mathbf{y} \quad (1)$$

kde predpokladáme plnú hodnotu matice  $\mathbf{A}$ . Všetky  $\mathbf{x}$ , ktoré splňujú  $\mathbf{Ax} = \mathbf{y}$  označujeme ako prípustné riešenia rovnice. Z predpokladu, že hľadáme čo najredší vektor  $\mathbf{x}$  vyplýva, že riešenie tejto rovnice je nekonečne mnoho.



Obrázok 1: [6] Schéma nedourčeného systému rovníc. Vektor  $\mathbf{x}$ , obsahuje prevažne nulové prvky a teda pozorovaný vektor  $\mathbf{y}$  je lineárna kombinácia iba pár stĺpcov matice  $\mathbf{A}$ . V obrázku je použitá pseudofarebná škála, kde modrá predstavuje nulové hodnoty a čím teplejšia farba, tým je hodnota vyššia

Vektor  $\mathbf{x} \in \mathbb{C}^N$  sa nazýva  $k$ -riedkym ( $k$ -sparse), ak platí

$$\|\mathbf{x}\|_0 \leq k \quad (2)$$

Takže  $k$ -riedky vektor je vektor, ktorý má najviac  $k$ -nenulových zložiek [6].

## 1.2 Podmienky pre jednoznačnosť riešenia

Aby bolo možné jednoznačne určiť, že riešenie daného problému je najredšie možné a zároveň jedinečné, potrebné uviesť nasledujúce vlastnosti.

### 1.2.1 Spark matice

Pre jednoznačnosť riešenia je potrebné zaviesť anglicky pojem spark, ktorý popisuje dôležité vlastnosti pri hľadaní riešenia [1]. Hodnota výrazu  $\text{spark}(\mathbf{A})$  sa definuje ako najmenší počet stĺpcov matice  $\mathbf{A}$ , ktoré sú lineárne závislé.

Pre nenulovú maticu  $\mathbf{A}$  pričom  $\mathbf{A} \in \mathbb{R}^{m \times N}$ , kde  $m < n$  platí, že  $\text{spark}$  môže nadobúdať hodnoty  $\text{spark}(\mathbf{A}) \in \{2, \dots, m + 1\}$ . Čím je menšia hodnota  $\text{spark}$ , tým redší musí byť vektor  $\mathbf{x}$ , aby bola zabezpečená jednoznačnosť riešenia [2,4].

Ak sústava  $\mathbf{A}\mathbf{x} = \mathbf{y}$  má riešenie splňujúce

$$\|\mathbf{x}\|_0 < \frac{\text{spark}(\mathbf{A})}{2} \quad (3)$$

tak  $\mathbf{x}$  je najredšie riešenie a žiadne iné s rovnakou riedkosťou neexistuje [2]. Nájdenie  $\text{spark}(\mathbf{A})$  je výpočtovo porovnateľné ako vyriešenie rovnice (1) a zároveň toto tvrdenie poskytuje iba dostačujúcu podmienku a nie nevyhnutnú, takže existujú aj riešenia, ktoré dané tvrdenie nespĺňujú [2].

### 1.2.2 Vzájomná koherencia (mutual coherence)

Vzájomná koherencia matice  $\mathbf{A}$  je definovaná ako najväčší absolútny normalizovaný skalárny súčin dvoch rôznych stĺpcov matice  $\mathbf{A}$

$$\mu(\mathbf{A}) = \max_{1 \leq j, k \leq N, j \neq k} \frac{|a_j^T a_k|}{\|a_j\|_2 \|a_k\|_2} \quad (4)$$

kde  $a_j$  označuje  $j$ -ty stĺpec matice  $\mathbf{A}$ . Vzájomná koherencia nám udáva mieru lineárnej závislosti medzi jednotlivými stĺpcami matice. Nedourčená sústava bude mať vždy nenulovú vzájomnú koherenciu [2].

Pokiaľ má sústava  $\mathbf{Ax} = \mathbf{y}$  riešenie  $\mathbf{x}$  splňujúce

$$\|\mathbf{x}\|_0 < \frac{1}{2} \left( 1 + \frac{1}{\mu(\mathbf{A})} \right) \quad (5)$$

tak potom  $\mathbf{x}$  je najredšie, aké je možné a zároveň jediné [2]. Toto riešenie sa dá dosiahnuť napr.  $l_1$ -minimalizáciou. Ak sa koherencia blíži k nule, pravá strana rastie nad všetky medze avšak ak sa koherencia blíži k jednej, tak aj pravá strana sa blíži k jednej. Snažíme sa hľadať a používať čo najviac nekoherentné slovníky [6].

### 1.2.3 Slovník vytvorený zlúčením dvoch ortobází (špeciálny prípad)

Ak spojíme dve ortonormálne bázy, tak nám vznikne slovník s  $2N$  stĺpcami a  $N$  riadkami. V literatúre sa uvádza, že sa dá dokázať, že pokiaľ  $\|\mathbf{x}\|_0 < 1/\mu(\mathbf{A})$ , tak  $\mathbf{x}$  musí byť najredšie možné riešenie. Podobne pokiaľ [4,6]

$$\|\mathbf{x}\|_0 < \frac{(\sqrt{2}-\frac{1}{2})}{\mu(\mathbf{A})} \quad (6)$$

tak  $\mathbf{x}$  je najredšie možné riešenie a súčasne, sa dá získať  $l_1$ -minimalizáciou [1].

## 1.3 Výpočtová zložitosť problému

Predpokladáme, že platí  $\text{spark}(\mathbf{A}) > 2k$  a existuje nejaké  $k$ -riedke riešenie sústavy, tak na základe tvrdenia 1.3 je toto riešenie najredšie a jednoznačné. Pokiaľ chceme nájsť presné riešenie, tak musíme prejsť všetkých  $\binom{N}{k}$  kombinácii podmnožín stĺpcov matice. V praxi, kde  $N$  je vysoké číslo, to je zdĺhavé. Ak chceme výsledok v rozumnom čase, tak je vhodné použiť aproximáciu [2].

## 1.4 Podmienky ekvivalencie riešenia $l_0$ - a $l_1$ -minimalizáciou

Motivácia pre dopracovanie sa k ekvivalentnému riešeniu alebo aspoň k podobnému je podrobnejšie rozpísaná v kap. 3.1.1. Pre dopracovanie sa k takémuto riešeniu je potrebné uviesť ďalšie definície.

### 1.4.1 NSP - vlastnosť nulového priestoru (Null Space Property)

Povedzme, že matica  $\mathbf{A} \in \mathbb{C}^{m \times N}$  splňuje NSP rádu  $k$  s konštantou  $\gamma \in (0,1)$ , pokiaľ platí

$$\|\boldsymbol{\eta}_T\|_1 \leq \gamma \|\boldsymbol{\eta}_{T^c}\|_1 \quad (7)$$

pre všetky množiny  $T \subset \{1, \dots, N\}$ ,  $|T| \leq k$  a pre všetky vektory  $\boldsymbol{\eta} \in \ker \mathbf{A}$ . Splnenie NSP zaisťuje pre ľubovoľný vektor z jadra matice  $\mathbf{A}$ , že v ňom bude norma koncentrovaná v malom počte prvkov. NSP zaisťuje, že  $k$ -riedke riešenie je jednoznačné a je možné ho nájsť  $l_1$ -minimalizáciou [4,5,6].

### 1.4.2 RIP - vlastnosť oslabenej izometrie (Restricted Isometry Property)

RIP v porovnaní s NSP je výpočtovo priateľskejšia alternatíva, ktorá je navyše stabilná aj pod vplyvom šumu.

Konštanta obmedzenej izometrie  $\delta_k$  matice  $\mathbf{A} \in \mathbb{C}^{m \times N}$  je najmenšie číslo, také, že platí

$$(1 - \delta_k) \leq \frac{\|\mathbf{A}\mathbf{z}\|_2}{\|\mathbf{z}\|_2} \leq (1 + \delta_k) \quad (8)$$

pre všetky vektory  $\mathbf{z} \in \Sigma_k^N$ . Povedzme, že matica  $\mathbf{A}$  spĺňa RIP rádu  $k$  s konštantou  $\delta_k$ , pokiaľ  $\delta_k \in (0,1)$  [4,5,6].

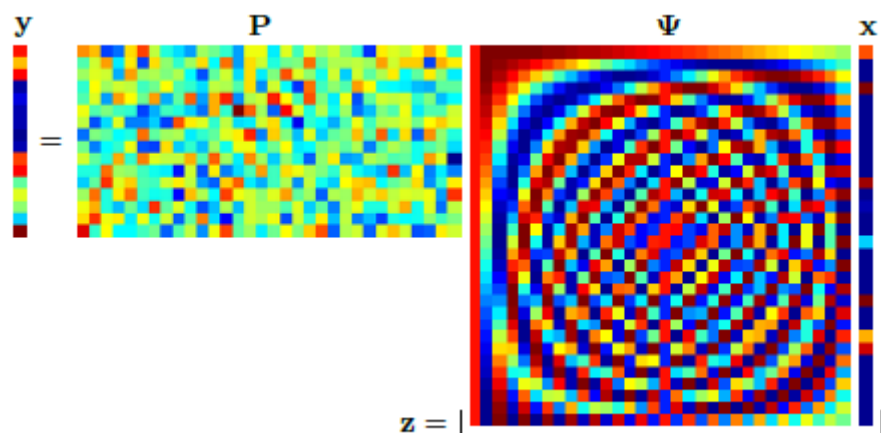
Pojem izometria znamená, že lineárne zobrazenie zachová dĺžku vektoru resp. je unitárne. Zoslabenie je dvojaké. Obmedzuje sa iba na všetky podmatice  $\mathbf{A}$  o  $k$  stĺpcoch a nepožadujeme presnú izometriu, ale je povolená malá odchýlka  $\delta_k$ . Do definície je nutné zahrnúť všetky podmatice o maximálne  $k$  stĺpcoch, pretože vo všeobecnosti nevieme, ktoré prvky vektoru  $\mathbf{x}$  budú nenulové a tým pádom nevieme, ktoré stĺpce matice  $\mathbf{A}$  sa podieľajú na signály  $\mathbf{y}$  [12].

## 2 Komprimované snímanie

Komprimované snímanie je jednou z dôležitých aplikácií v oblasti riedkych reprezentácií signálov. Základom celej teórie komprimovaného snímania, je predpoklad, že daný obraz je komprimovateľný resp. riedky v niektorých oblastiach transformácie [5,13].

Zvyčajný spôsob komprimovania prebieha až následne po nasnímaní všetkých dát. Najskôr nasnímame všetky dáta a až potom nasleduje proces komprimovania vhodnou transformáciou a následné vylúčenie zanedbateľných koeficientov pre ich malé nepodstatné hodnoty. Pri najpoužívanejšom kompresnom formáte JPEG, kde sú najskôr zosnímané pixely, potom prebehne dvojrozmerná diskretná kosínova transformácia (2D DCT) a následne kvantujeme koeficienty kosínovej transformácie. V ďalšom kroku nasleduje selekcia koeficientov, kde sa ponechajú koeficienty s najväčšou hodnotou. Proces v ktorom sa zakóduje iba ich pozícia a veľkosť prístupuje k signálu adaptívne.

Pri komprimovanom snímaní, za predpokladu, že je signál riedky pri vhodnej reprezentácii, v ktorejkoľvek časti transformácie, snímame signál lineárne a neadaptívne (bez ohľadu na jeho obsah). Počet nasnímaných vzoriek je potom omnoho menší ako počet vzoriek signálu. V tomto prípade sa zanedbáva Nyquistov vzorkovací teorém. To znamená, že signál snímame iba toľko krát, koľko je potrebné k jeho rekonštrukcii [3].



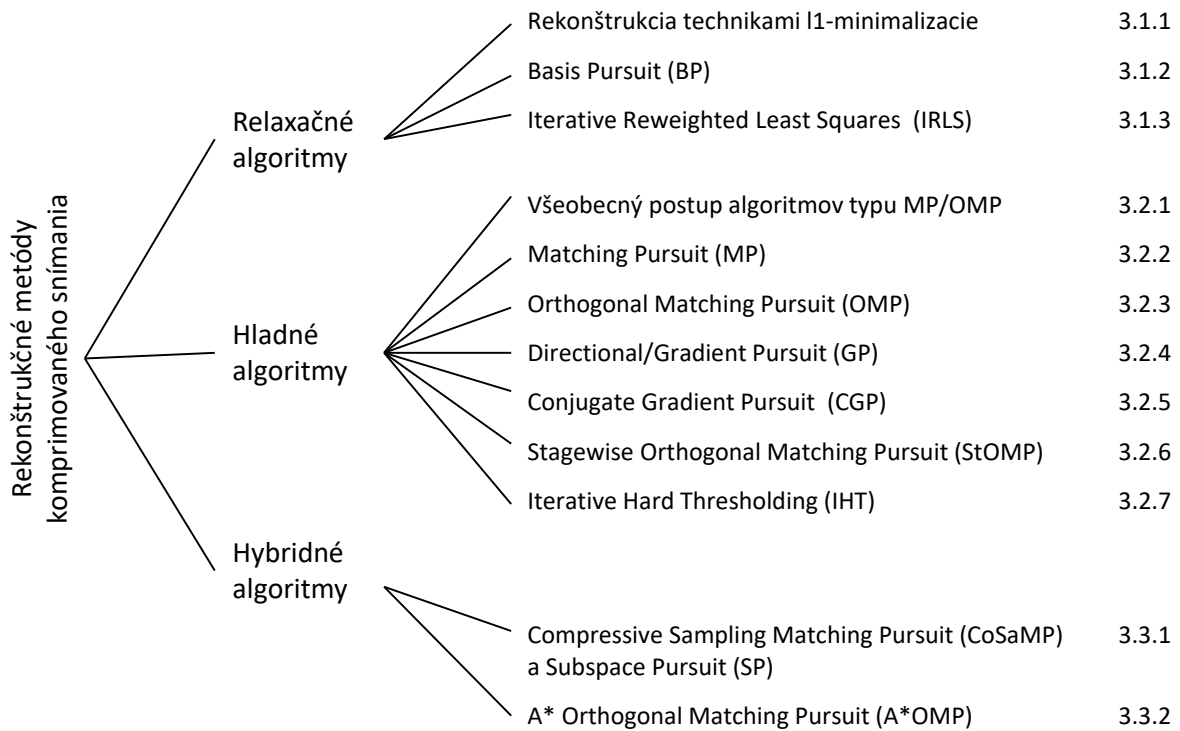
Obrázok 2: [7] Ilustračné schéma komprimovaného snímania. Nameraný vektor  $y$  je rovný súčinu meracej matice  $P$  a unitárnej matice  $\Psi$  a riedkeho vektoru  $x$ . Do procesu snímania vstupuje vektor  $z$ , ktorý je pozorovateľný a nie je sám o sebe riedky, ale je riedky v nejakej báze spätnej DCT. V obrázku je použitá pseudofarebná škála, kde modrá predstavuje nulové hodnoty a čím teplejšia farba, tým je hodnota vyššia.

Signál predpokladáme riedky v nejakom slovníku (ortogonálnej bazy). Bázu označíme  $\Psi$  a signál  $\mathbf{z}$  je vyjadrený ako  $\mathbf{z} = \Psi\mathbf{x}$ , kde  $\mathbf{x}$  je  $k$ -riedky. Cieľom je urobiť malý počet neadaptívnych meraní, ktoré budú mať charakter skalárneho súčinu so signálom, čo môžeme zapísať ako  $\mathbf{y} = \mathbf{P}\mathbf{z} = \mathbf{P}\Psi\mathbf{x}$ , kde  $\mathbf{P}$  je matica meraní typu  $m \times N$ , kde  $m \ll N$  a jednotlivé prvky vektoru  $\mathbf{y}$  sú výsledky merania, ktoré vznikajú ako lineárna kombinácia vzoriek signálu. Ilustrácie celej situácie je zobrazená na obr. 2. Rekonštrukcia signálu je už proces nelineárny a časovo náročnejší [7].



### 3 Rekonštrukčné metódy komprimovaného snímania

Ide o metódy riešenia úloh s riedkymi signálmi. Základne prístupy k nájdeniu riedkych riešení je možné rozdeliť do dvoch základných skupín. Prvá skupina vychádza z  $l_p$ -relaxácie a preto sa tieto metódy nazývajú relaxačné. Druhú skupinu tvoria tzv. hladné algoritmy (greedy pursuits) [6].



Obrázok 3: Prehľad rekonštrukčných metód komprimovaného snímania

### 3.1 Relaxačné algoritmy

Ako už bolo spomenuté a ako aj vyplýva z názvu, relaxačné algoritmy sú založené na  $l_p$  - relaxácii, ktorá nahradzuje  $l_0$ -normu. Tieto algoritmy spoliehajú na to, že za určitých podmienok nájdu buď presné alebo relatívne blízke riešenie. Do tejto skupiny patria: BP (Basis Pursuit), modifikovaná LARS (Least Angle Regression, homotopy method), IRLS (Iterative Reweighted Least Squares), ktorá sa často nazýva aj ako FOCUSS (FOCal Underdetermined System Solver) a ďalšie.[6,12]

#### 3.1.1 Rekonštrukcia technikami $l_1$ -minimalizácie

Keďže norma  $l_0$  nie je konvexná funkcia, tak sa rovnica (1) nedá riešiť žiadnou z metód konvexnej optimalizácie. Normy  $l_p$  sú však konvexné pre  $p \geq 1$  a preto sa dostávame k myšlienke, či by bolo možné využiť najbližšiu konvexnú normu  $l_1$  a dostať sa aspoň k približnému riešeniu problému, takže riešiť úlohu [6]

$$\min_x \|\mathbf{x}\|_1 \text{ vzhľadom k } \mathbf{Ax} = \mathbf{y} \quad (9)$$

V literatúre sa uvádza, že v niektorých prípadoch sa riešenie rovnice (1) a (9) skutočne zhodujú, dokonca takýchto prípadov je väčšina. Ak uvažujeme o dátach so šumom, tak je úloha formulovaná ako

$$\min_x \|\mathbf{x}\|_1 \text{ vzhľadom k } \|\mathbf{Ax} - \mathbf{y}\|_2 \leq \delta \quad (10)$$

kde  $\delta$  predstavuje odchýlku od presného riešenia.

Cieľom je nájsť riedke riešenie sústavy lineárnych rovníc, ako je popísané v rovnici (1). V praxi, ako už bolo vyššie uvedené, je nutné pre skrátenie výpočtového času použiť aproximáciu tejto rovnice a to buď pomocou tzv. hladných algoritmov (viz kap. 3.2) alebo pomocou  $l_1$ -minimalizácie

$$\min_x \|\mathbf{x}\|_1 \text{ vzhľadom k } \mathbf{Ax} = \mathbf{y} \quad (11)$$

Rekonštrukcia technikami  $l_1$ -minimalizácie prebieha nasledovne [7]

$$\mathbf{x}^1 := \arg \min \|\mathbf{x}\|_1 \text{ vzhľadom k } \mathbf{y} = \mathbf{P}\Psi\mathbf{x} \quad (12)$$

Ďalej bude uvedené za akých podmienok dosiahneme najredšie riešenie. Samotný signál  $\mathbf{z}$  potom dostaneme ako  $\mathbf{z} = \Psi\mathbf{x}^1$ . Vhodná meracia matica  $\mathbf{P}$ , tak aby bolo možné z malého počtu meraní naspäť zrekonštruovať signál, by mala byť v tvare  $\mathbf{P} = \mathbf{R}\Phi$ . Matica  $\Phi$  je matica (zatiaľ nešpecifikovaná) v tvare  $N \times N$  a  $\mathbf{R}$  je matica, ktorá vznikne z jednotkovej matice

o rozmeru  $N \times N$  ponechaním iba niekoľkých  $m$  náhodne vybraných riadkov a teda funguje ako náhodný vyber riadkov z  $\Phi$ . Náhodný výber z matice  $\mathbf{R}$ , sa realizuje rovnomerným rozdelením pravdepodobností. Maticu  $\mathbf{A}$  zapíšeme ako  $\mathbf{A} = \mathbf{R}\Phi\Psi$  a opäť sa dostávame k otázke ako zvoliť  $\Phi$  a  $m$ , aby matica  $\mathbf{A}$  umožňovala rekonštrukciu signálu  $l_1$ -technikami?

Existuje viacero možností. Definícia pre vzájomnú koherenciu je uvedená vyššie pre všeobecnú maticu. V špeciálnom prípade, keď je matica zložená z dvoch ortonormálnych bázy  $\Phi$  a  $\Psi$ , tak máme

$$\mu([\Phi, \Psi]) = \max_{1 \leq i, j \leq N} |\Psi_i^T \Phi_j| \quad (13)$$

Hodnota z rovnice (13) sa pohybuje medzi  $\frac{1}{\sqrt{N}}$  a 1 [4,7].

Ak je daný signál  $\mathbf{z}$ , ktorý má v  $\Psi$   $k$ -riedku reprezentáciu  $\mathbf{x}$ , potom riešenie  $l_1$ -minimalizácie (12), kde  $\mathbf{y}$  sú merania, je zároveň s vysokou pravdepodobnosťou najredšie možné riešenie, pokiaľ je dodržané [1]

$$m \geq C \cdot \mu^2([\Phi, \Psi]) \cdot k \cdot N \cdot \ln N \quad (14)$$

pre nejakú kladnú konštantu  $C$ . Z nerovnice nám vyplýva, že počet meraní závisí na riedkosti iba lineárne. Koherencia medzi ortobázami  $\Phi$  a  $\Psi$  kvadraticky ovplyvňuje nutný počet meraní. To vysvetľuje, prečo hľadáme dvojice s minimálnou koherenciou. Napríklad minimálnu koherenciu  $\frac{1}{\sqrt{N}}$  dosahuje dvojica  $[\mathbf{I}, \mathbf{F}]$ , kde  $\mathbf{I}$  je jednotková matica a  $\mathbf{F}$  je matica Fourierovej transformácie [1]. V takom prípade nám z nerovnice vyplýva, že počet meraní musí byť rádovo iba  $k \cdot \ln N$ . Naopak, ak koherencia rastie, tak na základe podmienky (14) prestáva byť meranie výhodné a dokonca v určitom momente prestane mať úplne zmysel, keďže  $m$  bude väčšie ako počet vzoriek  $N$ . To sa stane napríklad keď zvolíme  $\Phi = \Psi$ .

Ďalšia možnosť je pomocou RIP [5,7]. Matica  $\mathbf{P} = \mathbf{R}\Phi$ , kde  $\Phi$  je gaussovského a bernoulliovského typu, pričom to nie sú jediné možné konštrukcie. Vieme, že takéto matice  $\mathbf{P}$  s vysokou pravdepodobnosťou splňujú RIP ak  $m \geq C \cdot k \cdot \ln(N/k)$ . Gaussovské a bernoulliovské matice majú dôležitú vlastnosť, invarianciu proti unitárnej transformácii. Táto vlastnosť popisuje, že pokiaľ  $\mathbf{P}$  vyhovuje RIP, tak potom pre  $\mathbf{P}\Psi$  to platí taktiež a tým pádom aj celá matica  $\mathbf{A}$  vyhovuje RIP. Z tohto hľadiska sú uvedené matice univerzálne, lebo nezávisia na ortonormálnej báze, v ktorej je signál riedky [6,8].

Matica gaussovského typu - prvkami matice sú nezávislé hodnoty z normálneho rozdelenia so strednou hodnotou 0 a rozptylom  $1/m$  [5].

Matica bernoulliovského typu - každý prvok matice je náhodnou veličinou alternatívneho rozdelenia s hodnotou  $\pm 1/\sqrt{m}$  a s rovnakou pravdepodobnosťou výskytu [5].

### 3.1.2 Basis Pursuit (BP)

Algoritmus BP nahradzuje  $l_0$ -normu normou  $l_1$ , ktorá rieši rovnicu (10). Táto rovnica je ekvivalentná nasledujúcej rovnici [8]

$$\min_{\tilde{x}} 1^T \tilde{x} \text{ vzhľadom k } A\tilde{x} = y, \tilde{x} \geq 0 \quad (15)$$

Kde  $\tilde{x} = [x_+^T - x_-^T]$ , pričom  $x_+^T$  a  $x_-^T$  značí kladné resp. záporné koeficienty vektoru  $x$  a ostatné koeficienty sú vynulované. Touto rovnicou sa zaoberá lineárne programovanie a k jeho riešeniu existuje viacero algoritmov [7].

### 3.1.3 Iterative Reweighted Least Squares (IRLS)

Tento algoritmus sa často označuje aj ako FOCUSS (Focal Underdetermined System Solver). Využíva reprezentáciu  $l_p$ -normy ako váženú  $l_2$ -normu. Na začiatku sa musia nastaviť počiatočné podmienky a to aproximácia  $x = 1$  a základná váhová matica  $X = I$ , kde  $I$  predstavuje jednotkovú maticu. Následne sa v každej iterácii rieši v prvom kroku rovnica [11]

$$x = X^2 A^T (A X^2 A^T)^+ y \quad (16)$$

a v druhom kroku sa aktualizuje váhová matica  $X$

$$X(j, j) = |x(j)|^{1-p/2} \quad (17)$$

Algoritmus sa ukončí po splnení ukončovacej podmienky, ktorá je väčšinou nastavená ako rozdiel vektoru  $x$  v súčasnom a predchádzajúcom kroku v  $l_2$ -norme. [11]

## 3.2 Hladné algoritmy (Greedy pursuits)

Hladné algoritmy môžeme definovať ako súbor metód, ktoré na základe iterácii vytvárajú odhad  $x$ . Počínajúc nulovým vektorom na začiatku procesu, tieto metódy odhadujú množinu nenulových zložiek  $x$  iteračne, pričom pridávajú nové komponenty každou iteráciou. Tieto nové komponenty sa považujú za nenulové. Tento krok sa nazýva hladná selekcia (greedy selection/ element selection) a strieda sa s krokom odhadu, v ktorom sa hodnoty pre nenulové komponenty optimalizujú. Tieto metódy často vedú k veľmi rýchlym algoritmom, ktoré sa dajú aplikovať na veľmi veľké súbory dát. [4]

Výhodami tejto skupiny algoritmov sú výkonnosť, rýchlosť a malá zložitosť, pričom na druhej strane nevýhodou je, že nie je zaručené dosiahnutie vhodného globálneho optima riedkej aproximácie. Do tejto skupiny patria: MP (Matching Pursuit), OMP (Orthogonal Matching Pursuit), Thresholding algoritmy a ďalšie. [4,6,12]

### 3.2.1 Všeobecný postup algoritmov typu MP/OMP

Ako už bolo vyššie spomenuté, tieto hladné algoritmy môžeme označiť ako skupinu algoritmov, ktoré pracujú v dvoch základných krokoch a ukončovacej podmienke. Väčšinou sa na začiatku nastaví všeobecné počiatočné podmienky. Nulový odhad definujeme ako  $\hat{x}^{[0]} = 0$ . Na základe tejto inicializácie je možné zapísať zvyškovú chybu v tvare

$$r^{[0]} = y - A\hat{x}^{[0]} = y \quad (18)$$

a pomocnú sadu (tj. exponent pre nenulový prvok) pre prvý odhad  $\hat{x}^{[0]}$  je  $T = \emptyset$ . Pri každej iterácii sa tieto hodnoty menia vďaka pridaniu ďalších elementov (stĺpce matice  $A$ ) do pomocnej sady  $T$ . Z toho vyplýva, že po každej iterácii dostávame nový odhad signálu  $\hat{x}$  a tým pádom postupne znižuje pozorovanú zvyškovú chybu  $r$ . [4]

Tento postup sa v jednotlivých metódach trochu odlišuje. Najjednoduchšie metódy pridávajú iba skalárne násobky nového elementu k aktuálnej aproximácii, pričom viac sofistikované metódy prepočítávajú všetky prvky aktuálnej aproximácie. [12]

Po každej iterácii sa daný algoritmus musí rozhodnúť, či bude pokračovať tzn. či bude nasledovať ďalšia iterácia alebo nie. To sa určuje na základe ukončovacej podmienky. Najčastejšie sa ako podmienka používa počet iterácií, alebo chyba riedkej aproximácie. Ak sa používa chyba riedkej aproximácie  $r^{[i]}$ , tak sa proces ukončí ak bude chyba  $\|y - A\hat{x}^{[i]}\|_2^2$  menšia ako nastavený prah [4,12].

#### Všeobecný postup algoritmov typu MP/OMP

**Vstup:**  $y, A, k$

**for**  $i = 1 ; i := i + 1$ , pokiaľ nie je splnená ukončovacia podmienka

Počíta  $g^{[i]} = A^T r^{[i]}$  a vyberá elementy (stĺpce z  $A$ ) na základe hodnoty prvkov  $g^{[i]}$

Počíta a aktualizuje odhad  $\hat{x}^{[i]}$  (z neho aj  $\hat{y}^{[i]}$ ) znižovaním hodnoty funkcie

$$F(\hat{x}^{[i]}) = \|y - A\hat{x}^{[i]}\|_2^2$$

**end**

**Výstup:**  $r^{[i]}, x^{[i]}$

Obrázok 4: [4] Rámcové schéma všeobecného postupu hladných algoritmov

### 3.2.2 Matching Pursuit (MP)

Jeden z najjednoduchších algoritmov. Aproximácia je prírastková, vyberá jeden stĺpec z matice  $\mathbf{A}$  a pri každej iterácii sa mení iba koeficient spojený s daným stĺpcom. V každej iterácii sa aktualizuje

$$\hat{\mathbf{x}}_{j^{[i]}}^{[i]} = \frac{\hat{\mathbf{x}}_{j^{[i]}}^{[i-1]} + \mathbf{g}_{j^{[i]}}^{[i]}}{\|\mathbf{A}_{j^{[i]}}\|_2^2} \quad (19)$$

so snahou o minimalizáciu chyby riedkej aproximácie  $\|\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}^{[i]}\|_2^2$  na základe vybraného koeficientu.  $\mathbf{A}_j$  predstavuje  $j$ -ty stĺpec matice  $\mathbf{A}$ . Algoritmus MP bude opakovane vyberať tie isté stĺpce z matice  $\mathbf{A}$ , k nasledovnej úprave aproximácie, v ktorej  $\|\mathbf{r}\|^{[i]}$  konverguje lineárne k nule zakaždým pokiaľ stĺpce matice  $\mathbf{A} \in R^m$ .

MP sa ukončí v konečnom počte iterácií, ak je definovaná použitá norma  $\mathbf{r}^{[i]}$  ako ukončovacia podmienka pre algoritmus. MP vyžaduje opakované vyhodnocovanie násobenia matic, ktoré zahŕňa maticu  $\mathbf{A}^T$ , ktorá výrazne ovplyvňuje vypečenú náročnosť. Preto je výhodne používať túto metódu v prípadoch, kde nám je umožnená rýchla implementácia matice  $\mathbf{A}$  (často založená na FFT). [4]

#### Matching Pursuit (MP)

**Vstup:**  $y, A, k$

**for**  $i = 1 ; i := i + 1$ , pokiaľ nie je splnená ukončovacia podmienka

$$\mathbf{g}^{[i]} = \mathbf{A}^T \mathbf{r}^{[i-1]}$$

$$j^{[i]} = \operatorname{argmax}_j |\mathbf{g}_j^{[i]}| / \|\mathbf{A}_j\|_2$$

$$\hat{\mathbf{x}}_{j^{[i]}}^{[i]} = \hat{\mathbf{x}}_{j^{[i]}}^{[i-1]} + \mathbf{g}_{j^{[i]}}^{[i]} / \|\mathbf{A}_{j^{[i]}}\|_2^2$$

$$\mathbf{r}^{[i]} = \mathbf{r}^{[i-1]} - \mathbf{A}_{j^{[i]}} \mathbf{g}_{j^{[i]}}^{[i]} / \|\mathbf{A}_{j^{[i]}}\|_2^2$$

**end**

**Výstup:**  $\mathbf{r}^{[i]}, \hat{\mathbf{x}}^{[i]}$

Obrázok 5: [4] Rámcové schéma algoritmu Matching Pursuit (MP)

### 3.2.3 Orthogonal Matching Pursuit (OMP)

V tejto metóde je implementovaná sofistikovanejšia stratégia. V OMP je aproximácia pre  $x$  aktualizovaná v každej iterácii premietaním  $y$  ortogonálne na stĺpce matice  $A$  spojenými s pomocnou sadou  $T^{[i]}$ . OMP preto minimalizuje  $\|y - A\hat{x}^{[i]}\|_2^2$  cez všetky  $\hat{x}$  s pomocnou sadou  $T^{[i]}$ . V porovnaní s metódou MP, minimalizácia chyby je vykonávaná s ohľadom na všetky doteraz zvolené koeficienty [4,10]

$$\hat{x}_{T^{[i]}}^{[i]} = \underset{\tilde{x}_{T^{[i]}}}{\operatorname{argmin}} \|y - A_{T^{[i]}}\tilde{x}_{T^{[i]}}\|_2^2 \quad (20)$$

Na rozdiel od MP, OMP nikdy znovu nevyberá element a zvyšok v každej iterácii je vždy ortogonálny ku všetkým doposiaľ zvoleným elementom.

Pre všeobecné slovníky, rovnako ako aj pre MP, je výpočtová náročnosť OMP daná maticovo-vektorovými prvkami. Pri použití rýchlej transformácie je ortogonálny krok väčšinou prekážkou. Aj keď metóda OMP je výpočtovo komplexnejšia v porovnaní s metódou MP, tak vo všeobecnosti dosahuje vynikajúce výsledky, najmä v oblasti komprimovaného snímania. K riešeniu problému najmenších štvorcov sa používajú rôzne metódy [4,12].

#### Orthogonal Matching Pursuit (OMP)

**Vstup:**  $y, A, k$

$r^{[0]} = y, \hat{x}^{[0]} = 0, T^{[0]} = \emptyset$

**for**  $i = 1$ ;  $i := i + 1$ , pokiaľ nie je splnená ukončovacia podmienka

$g^{[i]} = A^T r^{[i-1]}$

$j^{[i]} = \operatorname{argmax}_j |g_j^{[i]}| / \|A_j\|_2$

$T^{[i]} = T^{[i-1]} \cup j^{[i]}$

$\hat{x}_{T^{[i]}}^{[i]} = A_{T^{[i]}} y$

$r^{[i]} = y - A\hat{x}^{[i]}$

**end**

**Výstup:**  $r^{[i]}, \hat{x}^{[i]}$

Obrázok 6: [4] Rámcové schéma algoritmu Orthogonal Matching Pursuit (OMP)

Existujú dva hlavné problémy pri aplikácii MP/OMP algoritmov na rozsiahle dáta. Prvým je výpočtová a pamäťová náročnosť pri jednej iterácii algoritmu. Druhým problémom je, výber jedného atómu. Pri tomto výber je potrebných presne  $k$ -iterácii na aproximáciu  $y$  s  $k$  atómami matice  $A$ . Ak má  $k$  veľkú hodnotu, tak tento proces môže byť až neprakticky pomalý [4].

Cieľom zavedenia smerových aktualizácií je získať aproximáciu ortogonálnej projekcie so zníženými výpočtovými nárokmi. Je to vlastne snaha o vyriešenie prvého problému.

### 3.2.4 Directional/Gradient Pursuit (GP)

Základnou črtou tohto algoritmu je jeho aktualizácia gradientu. Ako prirodzenú voľbu pre smer aktualizácie hľadáme záporný gradient funkcie  $\|\mathbf{y} - \mathbf{A}_{T^{[i]}}\tilde{\mathbf{x}}_{T^{[i]}}\|_2^2$  v rovnici

$$\mathbf{d}_{T^{[i]}}^{[i]} := \mathbf{g}_{T^{[i]}}^{[i]} = \mathbf{A}_{T^{[i]}}^T (\mathbf{y} - \mathbf{A}_{T^{[i]}}\hat{\mathbf{x}}_{T^{[i]}}^{[i-1]}) \quad (21)$$

Túto rovnicu nie je potrebné riešiť, pretože je to vlastne vedľajší produkt, ktorý nám vznikol z výberového procesu. Jednoducho obmedzíme vektor  $\mathbf{g}^{[i]}$ , ktorý už bol vypočítaný na prvky  $T^{[i]}$ . Použitím rovnice (21) sa dostávame k najzákladnejšej forme algoritmov so smerovou aktualizáciou [4,21].

Zvýšenie výpočtovej náročnosti oproti algoritmu MP je minimálne. Pritom, ak majú podmatice matice  $\mathbf{A}$  dobré vlastnosti (napríklad väčšiu hodnotu RIP), potom minimalizácia po smere gradientu môže poskytnúť dobrú aproximáciu k riešeniu problému najmenších štvorcov [4].

**Directorial Pursuit**

**Vstup:**  $y, A, k$

$r^{[0]} = y, \hat{\mathbf{x}}^{[0]} = 0, T^{[0]} = \emptyset$

**for**  $i = 1 ; i := i + 1$ , pokiaľ nie je splnená ukončovacia podmienka

$\mathbf{g}^{[i]} = \mathbf{A}^T r^{[i-1]}$

$j^{[i]} = \mathit{argmax}_j |g_j^{[i]}| / \|A_j\|_2$

$T^{[i]} = T^{[i-1]} \cup j^{[i]}$

počíta sa smer  $\mathbf{d}_{T^{[i]}}^{[i]}$  z  $\mathbf{c}^{[i]} = \mathbf{A}_{T^{[i]}} \mathbf{d}_{T^{[i]}}^{[i]}$  a  $\alpha^{[i]} = \frac{\langle r^{[i]}, \mathbf{c}^{[i]} \rangle}{\|\mathbf{c}^{[i]}\|_2^2}$

$\mathbf{x}_{T^{[i]}}^{[i]} := \mathbf{x}_{T^{[i]}}^{[i-1]} + \alpha^{[i]} \mathbf{d}_{T^{[i]}}^{[i]}$

$r^{[i]} = r^{[i-1]} - \alpha^{[i]} \mathbf{c}^{[i]}$

**end**

**Výstup:**  $r^{[i]}, \hat{\mathbf{x}}^{[i]}$

Obrázok 7: [4] Rámcové schéma algoritmu Directional/Gradient Pursuit

### 3.2.5 Conjugate Gradient Pursuit (CGP)

Algoritmus CGP má podobný princíp ako algoritmus GP, pričom vyžaduje o jeden maticovo-vektorový produkt  $\mathbf{A}_{T^{[i]}} \mathbf{g}_{T^{[i]}}^{[i]}$  navyše. Avšak ako sa uvádza v literatúre [19] na základe výpočtu tohoto produktu vieme priamo vyhodnotiť  $\mathbf{c}^{[i+1]}$ . Preto má algoritmus CGP presne takú istú výpočtovú náročnosť ako algoritmus GP. Okrem toho aktualizácia CGP bude vždy aspoň tak



dobrá ako aktualizácia GP, z hľadiska minimalizácie náročnosti funkcie, čo vo finále robí algoritmus CGP preferovanou možnosťou [19].

Ako už bolo vyššie spomenuté, druhým problémom s MP/OMP typmi stratégie je to, že je potreba vykonať aspoň taký počet iterácií, ktorý je zhodný s počtom vybraných atómov. Nasledujúce algoritmy sa snažia tento problém vyriešiť.

### 3.2.6 Stagewise Orthogonal Matching Pursuit (StOMP)

V klasických algoritmoch MP/OMP sa v kroku selekcie vyberá element, ktorý je maximálne korelovaný so zvyškom

$$j^{[i]} = \operatorname{argmax}_j \frac{|g_j^{[i]}|}{\|A_j\|_2} \quad (22)$$

Štandardnou stratégiou v stagewise algoritmoch je, že miesto maxima sa pre každú iteráciu nastaví prah  $\lambda^{[i]}$ , pričom jeho voľba sa v jednotlivých algoritmoch líši [17]. Algoritmus StOMP bol navrhnutý s cieľom zabezpečiť dostatočne výkonnú rekonštrukciu pri jeho použití v komprimovanom snímaní, pričom by sa mala udržať dostatočne nízka výpočtová náročnosť na to, aby tento algoritmus nemal problémy ani s rozsiahlymi dátovými súbormi. Práh sa určuje ako

$$\lambda_{stomp}^{[i]} = \frac{t^{[i]} \|r^{[i-1]}\|_2}{\sqrt{m}} \quad (23)$$

Pričom autori algoritmu tvrdia [17], že je vhodné voliť parameter  $t^{[i]}$  v rozmedzí  $2 < t^{[i]} < 3$ . Z praktického hľadiska sa javí, že výber parametru  $t$  jednoznačne ovplyvňuje výsledný výkon celého algoritmu. Špecifický problém, ktorý môže nastať pri týchto algoritmoch je ten, že sa môže predčasne ukončiť a to vtedy, ak všetky vnútorné produkty budú spadať pod prahovacu hranicu [4,17].

Pre lepší prehľad vo výpočtovej a pamäťovej náročnosti spomenutých algoritmov slúži tab.1.

Algoritmus	Výpočtová náročnosť	Pamäťová náročnosť
MP	$m + A + n$	$A + m + 2k + n$
OMP (QR)	$2mk + m + A + n$	$2(m + 1)k + 0.5k(k + 1) + A + n$
OMP (Chol) <sup>1</sup>	$3A + 3k^2 + 2m + n$	$0.5k(i + 1) + A + m + 2k + n$
GP	$2A + k + 3m + n$	$2m + A + 2k + n$
CGP	$2A + k + 3m + n$	$2m + A + 2k + n$
StOMP	$(v + 2)A + k + 3m + n$	$2m + A + 2k + n$

Tabuľka 1: [4] Prehľad výpočtovej a pamäťovej náročnosti pre jednu iteráciu

<sup>1</sup> K riešeniu problému najmenších štvorcov boli použité dva druhy faktorizácie, QR faktorizácia a Cholského faktorizácia, viac informácií v literatúre [4]

### Prahovacie algoritmy (Thresholding type algorithms)

Ako bolo vyššie uvedené, hladné algoritmy sa dajú ľahko implementovať, používať a môžu byť extrémne rýchle. Avšak nemajú záruky rekonštrukcie tak silné ako metódy založené na konvexnej relaxácii. Metóda popísaná v tejto časti dba práve na tento nedostatok. Môže byť extrémne rýchle a poskytujú výborné záruky rekonštrukcie. [4]

#### 3.2.7 Iterative Hard Thresholding (IHT)

IHT je hladný algoritmus, ktorý iteratívne rieši lokálnu aproximáciu pri rekonštrukcii signálu komprimovaného snímania

$$\min_{\tilde{\mathbf{x}}} \|\mathbf{y} - \mathbf{A}\tilde{\mathbf{x}}\|_2^2 \text{ vzhľadom k } \|\tilde{\mathbf{x}}\|_0 \leq k \quad (29)$$

Po úprave na základe literatúry [4], sa dostaneme k rovnici

$$\hat{\mathbf{x}} = H_k(\mathbf{z} + \mu\mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{z})) \quad (30)$$

kde  $H_k$  je nelineárne zobrazenie, ktoré nastaví všetky elementy, okrem  $k$ -najväčších koeficientov na nulu. V prípade, že  $k$  najväčších koeficientov nie je jednoznačne zadefinovaných, tak predpokladáme, že si algoritmus vyberie z koeficientov na základe preddefinovaného poradia. Táto lokálna optimalizácia sa začlení do iteratívneho algoritmu nastavením  $\mathbf{z} = \hat{\mathbf{x}}^i$ . Na základe aktuálneho odhadu  $\hat{\mathbf{x}}^{[i]}$  algoritmus hladovo hľadá globálne minimum [4].

Algoritmus IHT je možno pomerne jednoducho implementovať a je výpočtovo efektívny. Jeho hlavnými výpočtovými krokmi okrem vektorových dodatkov sú násobenia vektorov s maticou  $\mathbf{A}$  a jej transponovanou, ako aj čiastočné triedenie potrebné pre krok prahovania. Pamäťové nároky sú taktiež malé a ak sú použité štruktúrované meracie matice, tak aj násobenie s maticou  $\mathbf{A}$  a jej transponovanou môže byť vykonané efektívne [4].

#### Iterative Hard Thresholding algorithm (IHT)

**Vstup:**  $y, A, k$

$\hat{\mathbf{x}}^{[0]} = 0$

**for**  $i = 0, i := i + 1$  pokiaľ nie je splnená ukončovacia podmienka

$\hat{\mathbf{x}}^{[i+1]} = H_k(\hat{\mathbf{x}}^{[i]} + \mu\mathbf{A}^T(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}^{[i]}))$

**end**

**Výstup:**  $\hat{\mathbf{x}}^{[i]}$

Obrázok 8: [4] Rámcové schéma algoritmu Iterative Hard Thresholding

### 3.3 Hybridné algoritmy

Hybridné algoritmy využívajú popredné vlastnosti oboch skupín algoritmov. Do tejto skupiny patria napríklad algoritmy pre rôzne aplikácie napr. Compressive Sampling Matching Pursuit (CoSaMP), Subspace Pursuit (SP), A\* Orthogonal Matching Pursuit (A\*OMP) a iné. [4,6]

#### 3.3.1 Compressive Sampling Matching Pursuit (CoSaMP) a Subspace Pursuit (SP)

Oba algoritmy sledujú aktívnu sadu  $T$  nenulových elementov a oba ako aj pridávajú, tak aj odoberajú elementy v každej iterácii. Na začiatku každej iterácie, je  $k$ -riedky odhad  $\hat{x}^{[i]}$  použitý na výpočet zvyškovej chyby  $y - A\hat{x}^{[i]}$ , ktorého vnútorné prvky sú počítane spolu so stĺpcami matice  $A$ . Indexy týchto stĺpcov matice  $A$  s  $k$  (alebo  $2k$ ) najväčšími prvkami vnútorného produktu sú vybrané a pridané do pomocnej sady odhadu na získanie väčšej sady  $T^{[i+0,5]}$ . Dočasný odhad  $\hat{x}^{[i+0,5]}$  je potom počítaný pomocou metódy najmenších štvorcov zo vzťahu

$$\operatorname{argmin}_{\tilde{x}_{T^{[i+0,5]}}} \left\| y - A \tilde{x}_{T^{[i+0,5]}} \right\|_2 \quad (31)$$

Hľadá sa  $k$ -najväčších elementov tohto dočasného odhadu, ktoré vytvoria novú pomocnú sadu  $T^{[i+1]}$  [4].

Metódy sa líšia až v poslednom kroku, kde algoritmus CoSaMP zoberie nový odhad dočasného odhadu  $\hat{x}^{[i+0,5]}$  vzhľadom k novej menšej pomocnej sade  $T^{[i+1]}$ , zatiaľ čo algoritmus SP rieši druhú metódou najmenších štvorcov vzhľadom k zredukovanej pomocnej sade [4].

Hlavný rozdiel medzi týmito dvoma algoritmi je vo veľkosti sady, ktorá sa pridáva  $T^{[i]}$  v každej iterácii, ako aj ďalšie potrebné riešenie metódy najmenších štvorcov pri algoritme SP. Z možnosti nahradiť riešenie metódy najmenších štvorcov s tromi aktualizáciami na základe gradientu vyplýva, že algoritmus CoSaMP sa dá implementovať efektívnejšie ako algoritmus SP [4].

#### 3.3.2 A\* Orthogonal Matching Pursuit (A\*OMP)

Tento algoritmus používa A\*-algoritmy na prehľadávanie informačných stromov. Hodnotiacia funkcia ohodnocuje a označuje jednotlivé cesty a samotná cesta je počítaná pomocou OMP. Algoritmus A\* pracuje vo viacerých stromoch paralelne, pričom vyhľadáva najviac perspektívnu cestu v strome a rozširuje ju pridávaním nových uzlov (elementov). V ideálnom prípade by sa vždy pridalo k danému bodu  $N - 1$  uzlov, čo by však nebolo veľmi efektívne a preto sa využíva algoritmus OMP. Pri hľadaní  $k$ -riedkeho riešenia sa hľadá optimálna cesta s  $k$  uzlami. Pri tomto algoritme nie je jednoduché určenie počiatkových podmienok, pretože sa musí určiť koľko elementov sa použije ako vrchol stromu, čo nám určuje koľko stromov sa bude počítať [9,12].

## 4 Aplikácie komprimovaného snímania

Aplikácia komprimovaného snímania si postupne nachádza uplatnenie pri snímaní a spracovaní obrazu [13]. V tejto kapitole sú uvedené príklady, kde je možné využiť komprimované snímanie v oblasti multimedialnej techniky. Podrobnejší popis sa nachádza pri jednopixelovej kamere v kap. 4.3, ktorej model bol predmetom návrhu a implementácie v praktickej časti.

### 4.1 Potlačovanie šumu (denoising)

Hovoríme o klasickom prípade, kde ideálny obraz  $x$  obsahuje biely Gaussovský šum  $v$  so štandardnou odchýlkou  $\sigma$ . Pre výsledný obraz platí  $y = x + v$ . Cieľom je navrhnúť algoritmus, ktorý odstráni šum z  $y$  a dopracovať sa čo najbližšie k pôvodnému  $x$ . Tento prístup je založený na riedkej a redundantnej reprezentácii natrénovaných slovníkov [13,15].

Algoritmus K-SVD slúži na natréovanie slovníkov. Je flexibilný a funguje v spojení s akýmkoľvek druhom "pursuit" algoritmov. Algoritmus K-SVD je podrobne popísaný v literatúre [24]. Použitím tohto algoritmu získame slovník, ktorý efektívne popisuje obrázok. Na natréovanie slovníka sa používajú dve možnosti. Prvou je použitie samotného nedokonalého obrázka z ktorého získame redundantný DCT slovník. Druhou je, že získame korpus z vysoko kvalitnej obrázkovej databázy. K-SVD algoritmus má obmedzené využitie pri práci s malými (lokálnymi) časťami obrazu. Preto pred tým, ako sa dopracujeme k riedkosti v jednotlivých častiach obrazu definujeme globálny obraz, na ktorý následne môžeme použiť tento algoritmus. S každou iteráciou, ktorá zahrňuje kroky odšumovania obrázka je spojená aj aktualizácia slovníku. Použitie Bayesiankej metódy vedie k jednoduchému a efektívnemu odšumovaciemu algoritmu [15].

Taktiež je možné riešiť tento problém aj iným spôsobom, ako je uvedené v literatúre [13]. Kde na miesto toho, aby sa problém riešil ako celok (globálne), tak sa obraz rozdelí na  $p$ -malých častí, ktoré sú doostrované samostatne a nezávisle na sebe, takže sa pracuje v lokálnej škále [13].

### 4.2 Doostrovanie (deblurring)

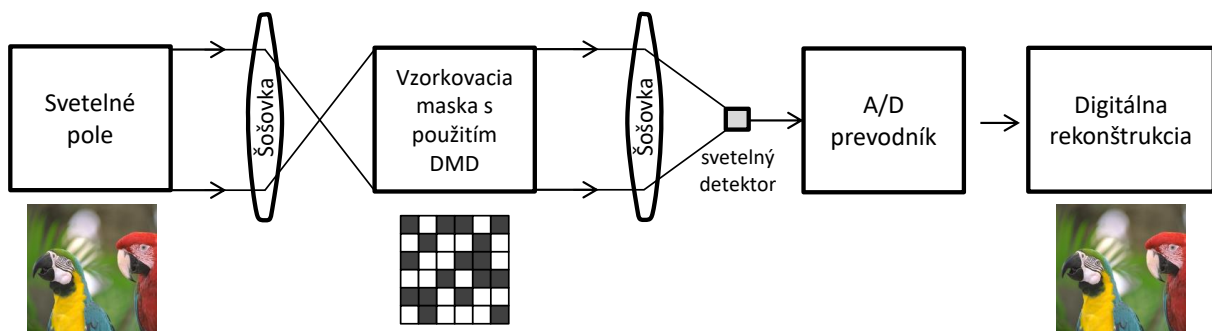
Predpokladáme, že máme originálny obraz  $y$  o veľkosti  $\sqrt{n} \times \sqrt{n}$  pixelov, ktorý prejde známou lineárne priestorovo invariantnou operáciou rozmazania  $H$  a následne je pridaný biely Gaussovský šum. Vo výsledku máme rozmazaný obraz so šumom o rovnakej veľkosti  $\tilde{y} = Hy + v$ , kde sa snažíme odhadnúť pôvodný obraz  $y$  za predpokladu, že sa dá popísať ako  $Ax$ , kde  $A$  je známy slovník a  $x$  je riedky vektor [13]

$$\hat{x} = \operatorname{argmin}_x \frac{1}{2} \|\tilde{y} - HAx\|_2^2 + \lambda \mathbf{1}^T \rho(x) \quad (32)$$

kde  $\rho(\mathbf{x})$  sa mení pri každom novom vstupe  $\mathbf{x}$ . Slovník  $\mathbf{A}$  väčšinou využíva výber  $\rho(\mathbf{x})$ , kde  $\rho(\mathbf{x}) = |\mathbf{x}|^p$  a  $p \leq 1$  čo vedie k  $l_p$ - norme. Obnovený obraz je daný ako  $\hat{\mathbf{y}} = \mathbf{A}\hat{\mathbf{x}}$  a cieľom je sa čo najviac priblížiť k originálnemu obrazu vzhľadom k MSE(mean-squared-error). Aby rovnica (32) viedla k praktickému algoritmu na doostrenie obrazu musíme špecifikovať slovník  $\mathbf{A}$  tak, aby sme sa dopracovali k riedkemu reprezentačnému modelu obrazového obsahu. V literatúre [13] sa uvádza využitie translačno-invariantnej vlnkovej transformácie, ktorá využíva Haar filter s dvomi vrstvami rozlíšenia, čo vedie k redundantnému faktoru 7:1. Následne dostaneme 7 obrazov o veľkosti  $\sqrt{n} \times \sqrt{n}$  pixelov. Tento celý proces sa dá popísať ako  $\mathbf{A}^T \mathbf{y}$ . Takto zvolený slovník je ďaleko od optimálnej voľby pre riedku reprezentáciu obrazov, pričom lepšie výsledky môžu byť dosiahnuté, ak sa použije adekvátnejší slovník. Operácia  $\mathbf{A}\mathbf{x}$  je vykonaná inverznou Haarovou transformáciou pre tento redundantný prípad. Použitím inverznej transformácie na vektor  $\mathbf{x} = \mathbf{e}_i$  (kde iba  $i$ -ty vstup sa rovná jednej a všetky ostatné sa rovnajú nule) udáva  $i$ -ty stĺpec matice  $\mathbf{A}$ , čo môže byť použité pre vizualizáciu atómov v tomto slovníku. Ako už bolo spomenuté na základe voľby nie úplne adekvátneho slovníka, táto spomenutá metóda nie je dokonalá, avšak dosahuje postačujúce výsledky [13].

### 4.3 Jednapixelová CS kamera

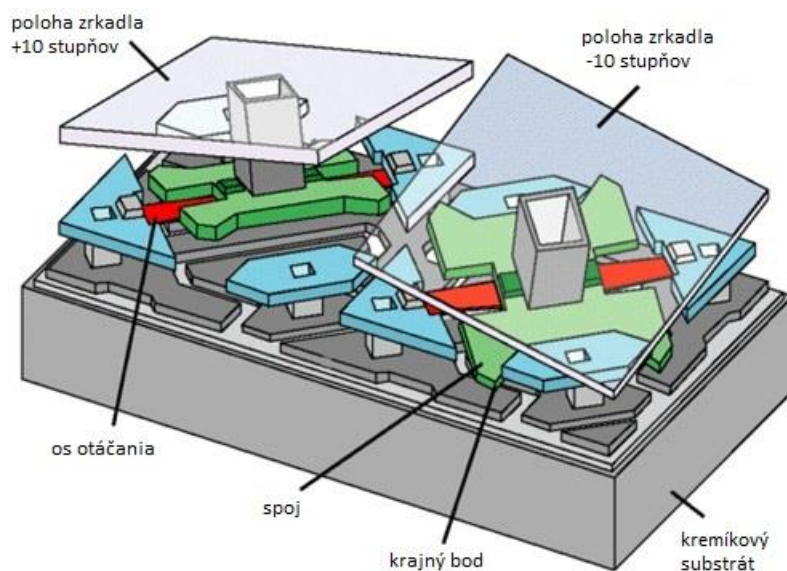
Štandardná kamera má veľké pole fotónových detektorov, jeden odpovedajúci pre každý pixel obrazu. V súčasnej dobe ich sú milióny. Na rozdiel od nej, sa jednapixelová kamera nazýva jednapixelovou z toho dôvodu, že má len jeden jediný fotónový detektor. Fotónové detektory pre netradičné vlnové pásma, môžu byť oveľa rozsiahlejšie a drahšie na výrobu. To by mohla byť napríklad motivácia pre uplatnenie jednapixelovej kamery. To čo však najviac vyzdvihuje jednapixelovú CS kameru, je jej prístup vo vzorkovaní, ktorý vychádza teórie komprimovaného snímania. Na základe tejto teórie nám postačí na rekonštrukciu obrazu omnoho menej meraní ako je celkový počet pixelov vstupného obrazu [14]. Ďalšie výhody (viz kap. 4.3.1), ako aj nevýhody (viz kap. 4.3.2) sú podrobne popísané ďalej v texte.



Obrázok 9: [16] Schéma jednapixelovej CS kamery

Schéma jednopixelovej kamery je vyobrazené na obr. 9. Dopadajúce svetelné pole je zamerané na špeciálny typ priestorového svetelného modulátora (SLM - Spatial Light Modulator), ktorý väčšinou pozostáva z kremíkového čipu s veľkým množstvom malých zrkadiel (DMD - Digital Micromirror Device), z ktorých každé jedno zrkadlo v ideálnom prípade zodpovedá jednému pixelu, to však záleží na rozlíšení vstupného obrazu.

Každé zrkadlo môžeme nastaviť do dvoch rôznych polôh. Poloha "on" (nastavenie zrkadla do polohy  $+10^\circ$ ) smeruje svetlo pre daný pixel smerom k detektoru, zatiaľ čo poloha "off" (nastavenie zrkadla do polohy  $-10^\circ$ ) smeruje svetlo mimo detektor. Ukážkové nastavenie zrkadiel sa nachádza na obr. 10. Svetlo zo všetkých pixelov nastavených v polohe "on" je zhromaždené pomocou šošovky a zamerané na jeden fotónový detektor (fotodiódu), z ktorého dostávame ako výsledok merania výstupne napätie. Toto napätie je potom digitalizované pomocou A/D prevodníka. [16]



Obrázok 10: [22] Schéma dvoch zrkadiel DMD zariadenia

Meranie sa následne zopakuje určitý počet krát, pričom zakaždým sú polohy zrkadiel náhodne nastavené. Výsledkom takéhoto procesu je obraz zakódovaný vo veľkom množstve hodnôt výstupného napätia z fotónového detektoru. Najväčšou výzvou v tomto prístupe jednopixelového snímania je správne dekódovanie výstupných dát a ich následná rekonštrukcia na obraz pomocou metód komprimovaného snímania (viz kap. 3) [14,16].

Koncepcia jednopixelovej kamery nie je nová a patrí do širokej kategórie multiplexných snímacích metód, v ktorých sa vykonáva séria postupných meraní jediným detektorom. To čo však vyzdvihuje jednopixelovú CS kameru, je nový prístup vo vzorkovaní na základe teórie komprimovaného snímania [14]. Ten nám hovorí, že nám postačuje omnoho menší počet meraní ako je celkový počet pixelov vstupného obrazu. Jedná sa o alternatívny prístup k

získaniu komprimovaných obrázkov. Namiesto toho, aby sme nasníмали všetky vzorky obrazu a potom ich následne zkomprimovali, tak sa snažíme už priamo nasnímať komprimované vzorky. Aby sme toho dosiahli, teória komprimovaného snímania nás vedie k použitiu náhodného postupu odberu vzoriek, pri ktorom je každé zrkadlo v DMD zariadení nastavené náhodne do polohy “on“ alebo “off“ s rovnakou pravdepodobnosťou. [16]

#### **4.3.1 Výhody jednopixelovej kamery**

Rovnako ako aj pri iných architektúrach s jedným pixelom je potrebný iba jeden jediný fotónový detektor vzhľadom k tradičnému prístupu snímania, kde sú ich v súčasnej dobe milióny. V porovnaní s viditeľným svetlom môžu byť fotónové detektory pre iné vlnové pásma oveľa rozsiahlejšie a drahšie na výrobu. Tento jednopixelový prístup by preto mohol umožniť získanie obrazu na vlnových dĺžkach, pre ktoré by to inak nebolo možné z dôvodu veľkosti alebo cenových obmedzení. [14]

Na základe potreby menšieho počtu vzoriek by sa dalo povedať, že čas snímania vzoriek by mohol byť kratší v porovnaní s inými jednopixelovými prístupmi. To by mohlo byť dôležité napríklad pre využitie v systémoch, ktoré potrebujú rýchlo reagovať. [16]

Pretože sa odoberá menej vzoriek, objem získaných dát sa taktiež zníži, čo je ďalšia dôležitá úloha v dnešnej dobe, pretože sa často vyskytujú obmedzenia v objeme dát, ktoré môžu byť uložené alebo prenášané. V tradičných metódach snímania, kde musíme najskôr nasnímať všetky vzorky a až potom ich následne komprimujeme, prenášame a pracujeme s veľkým množstvom dát. Pri jednopixelovej kamere je výpočtovo náročná až fáza rekonštrukcie obrazu. To by sa dalo využiť napríklad pri vyhotovovaní družicových záberov. [16]

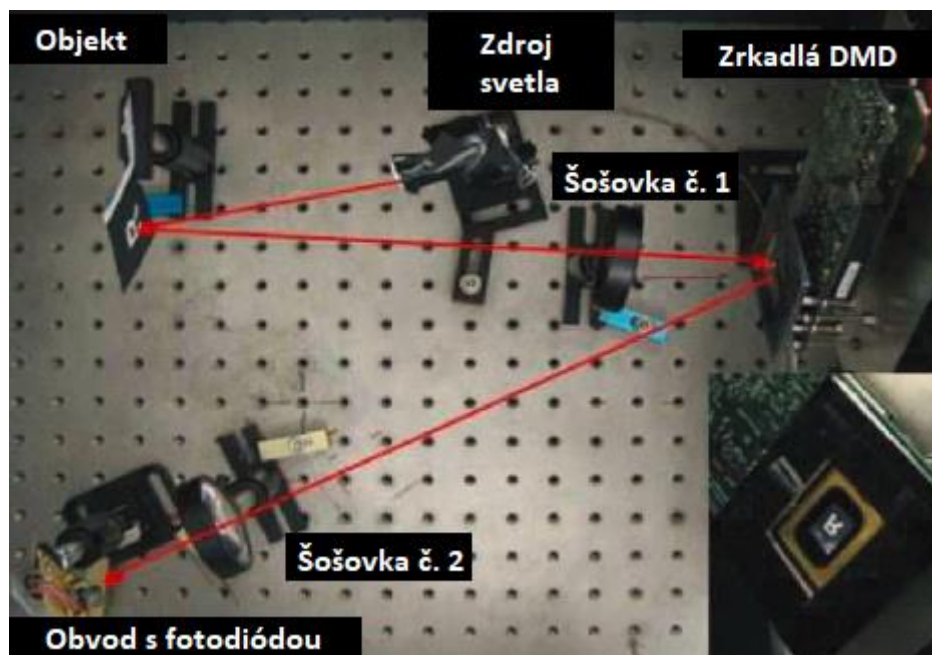
Medzi ďalšie praktické výhody jednopixelového dizajnu okrem flexibility snímania patrí aj skutočnosť, že kvantová účinnosť fotodiódy je vyššia ako tá u pixelových snímačov v typickom CCD alebo CMOS a faktor plnenia DMD môže dosiahnuť až 90%, zatiaľ čo pri CCD/CMOS má iba okolo 50%. Ďalšou dôležitou výhodou je fakt, že každé meranie CS prijme približne  $N/2$  krát viac fotónov ako priemerný pixelový senzor, čo výrazne znižuje skreslenie obrazu napríklad od čierneho šumu. Ďalšími teoretickými výhodami je jeho univerzálnosť, robustnosť a progresivita [14].

#### **4.3.2 Nevýhody jednopixelovej kamery**

Medzi najväčšie výzvy pri využívaní konceptu jednopixelovej kamery patrí určite proces rekonštrukcie obrazu. Do dnešného dňa bolo navrhnuté veľké množstvo algoritmov na rekonštrukciu riedkych signálov, z ktorých si väčšina nájde uplatnenie v komprimovanom snímaní. Jednoznačne sa nedá určiť, ktorý algoritmus je najlepší na rekonštrukciu obrazov. Tento proces rekonštrukcie ovplyvňuje veľké množstvo faktorov, ako aj z hľadiska vstupného obrazu, tak aj v nastavovaní rozličných parametrov daného algoritmu. [23]

### 4.3.3 Matematický popis jednopixelovej kamery

Jednopanelová kamera je optický počítač, ktorý sekvenčne meria vnútorné produkty  $\mathbf{y}[m] = \langle \mathbf{x}, \Phi_m \rangle$ , medzi N-pixelovou vzorkovanou verziou dopadajúceho svetelného poľa zo scény  $\mathbf{x}$  a sadou 2D testovacích funkcií  $[\Phi_m]$ . Ako je znázornené na obr. 11 svetelné pole je zamerané bikonvexnou šošovkou na DMD pozostávajúce z poľa N drobných zrkadiel. Každé zrkadlo odpovedá určitému pixelu v  $\mathbf{x}$  a  $\Phi_m$  a môže byť nezávisle orientované buď smerom k šošovke č.2 (čo predstavuje hodnotu 1 pre pixel v  $\Phi_m$ ) alebo smerom od šošovky č.2 (čo predstavuje hodnotu 0 pre pixel v  $\Phi_m$ ). Odrazené svetlo je potom zhromaždené pomocou bikonvexnej šošovky a zamerané na jediný fotónový detektor (jediný pixel), ktorý integruje produkt  $x[n]\Phi_m[n]$  na výpočet merania  $\mathbf{y}[m] = \langle \mathbf{x}, \Phi_m \rangle$  ako jeho výstupne napätie. Toto napätie je potom digitalizované A/D prevodníkom. Hodnoty  $\Phi_m$  medzi nulou a jednotkou môžeme získať chvením zrkadiel počas integračného času fotodiódy. Na získanie kladných a záporných hodnôt pre  $\Phi_m$  odhadujeme a odčítame strednú hodnotu intenzity svetla z každého merania, ktorá sa dá ľahko získať nastavením úplne všetkých zrkadiel na jednu pozíciu. Na výpočet náhodných meraní CS, kde  $\mathbf{y} = \Phi\mathbf{x}$ , náhodne nastavíme orientáciu zrkadiel  $\Phi_m$  pomocou pseudonáhodného generátora čísel, odmeriame  $\mathbf{y}[m]$  a proces opakujeme M-krát, aby sme získali vektor  $\mathbf{y}$  [14].



Obrázok 11: [14] Príklad rozmiestnenia hardwaru jednopixelovej kamery



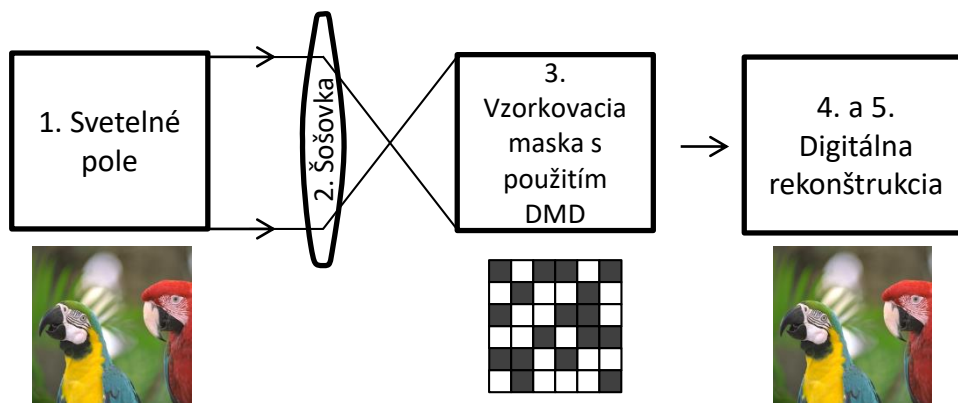
## 5 Praktická časť

Cieľom praktickej časti práce bolo navrhnuť a implementovať simuláciu modelu jednopixelovej kamery. Motiváciou, pre výber práve jednopixelovej kamery bolo hneď niekoľko. Či už ako je popísané v úvode, snaha vytvoriť kameru, ktorá nepotrebuje na kvalitný obraz milióny fotónových detektorov, alebo jej ďalšia rada výhod (viz kap. 4.3.1), ktorými taktiež disponuje.

Tento model môže byť rozdelený do dvoch základných častí, časť vzorkovania (spracovania) po ktorej nasleduje časť rekonštrukcie. V prvej časti je snímaný vstupný obraz, na ktorý sa následne aplikuje vzorkovacia maska. V druhej časti to je proces rekonštrukcie, ktorý využíva algoritmus pracujúci na teórii komprimovaného snímania. Na implementáciu modelu bol použitý simulačný program MATLAB.

### 5.1 Popis modelu

Model, ktorý bol použitý v simulácii (viz obr. 12) je mierne zjednodušený oproti modelu, ktorý je uvedený v predchádzajúcej kapitole (viz kap. 4). Neobsahuje simuláciu fotodiódy.



Obrázok 12: Schéma simulácie jednopixelovej kamery

Simulácia modelu jednopixelovej kamery sa tým pádom skladá z nasledujúcich častí:

#### 1. Vstupný obraz

Do simulácie je možné vložiť ľubovoľný vstupný obraz, nezáleží na veľkosti a ani na tom, či je obraz farebný alebo šedotónový.

## 2. Snímací objektiv

V simulácii je taktiež potrebné zohľadniť aj optickú časť. Prechod cez šošovku je preto simulovaný ako prechod cez objektiv, ktorý má gaussovskú rozptylovú funkciu (PSF).

## 3. Vzorkovacia maska

V tejto časti prebieha aplikácia DMD modelu, ktorý sa skladá z veľkého množstva malinkých zrkadiel (viz kap. 4.3). V simulácii to odpovedá náhodne vygenerovanej matici, ktorá obsahuje hodnoty "0" a "1".

## 4. Bloková transformácia

Základom celej teórie komprimovaného snímania, je predpoklad, že daný obraz je komprimovateľný resp. riedky v niektorých oblastiach transformácie (viz kap. 2). V simulácii je použitá najrozšírenejšia bloková transformácia, ktorá je základom kompresného štandardu JPEG, diskretnú kosínovú transformáciu (DCT). Táto transformácia prebieha pre každý blok obrazu zvlášť a nezávisle. Na základe tejto transformácie sa predpokladá, že obraz je dobre komprimovateľný teda riedky.

## 5. Rekonštrukčný algoritmus

V simulácii je použitý jeden zo základných algoritmov, algoritmus OMP. Jeho teoretický rozbor ako aj blokové schéma je popísané vyššie v texte, v kapitole 3.2.3. Tento algoritmus bol vybraný pre jeho dobrý pomer medzi výpočtovou náročnosťou a výslednou kvalitou rekonštruovaného obrazu. Okrem toho táto metóda je najrozšírenejšia a najpoužívanejšia, preto ponúka široké uplatnenie v praxi.

Ako bolo už spomenuté v prvej časti modelu, pri vstupnom obraze nezáleží na tom či je farebný alebo iba šedotónový. Simulácia funguje pri oboch farebných rozlíšeniach. Pokiaľ ide o farebný obraz, tak rekonštrukcia prebieha postupne po jednotlivých farebných zložkách (červená, zelená a modrá) zvlášť, avšak na rovnakom princípe ako pre šedotónový obraz.

## 6 Vyhodnotenie výsledkov

V nasledujúcich podkapitolách sú podrobne popísané typy simulácií, ktoré boli prevedené. V prvej boli skúmané rekonštrukčné kvality algoritmu OMP. V nasledujúcich boli skúmané simulácie jednopixelovej kamery na obrazy so štandardným rozlíšením ako aj na obrazy s vysokým rozlíšením. S výnimkou simulácií s obrazmi s vysokým rozlíšením boli na simulácie použité obrazy z voľne dostupnej databázy Kodaku<sup>2</sup>, ktoré majú štandardné rozlíšenie 768x512 resp. 512x768 pixelov.

Všetky simulácie boli odsimulované v programovacom prostredí MATLAB na notebooku s procesorom Intel Core i5 2520M s frekvenciou 2,50 GHz a pamäťou RAM 4 GB. Na vyhodnocovanie výsledkov dosiahnutej kvality obrazu sú použité objektívne metriky na meranie kvality obrazu PSNR a SSIM. Dosiahnutá hodnota metriky je v grafoch vynesená (na ose  $y$ ) vzhľadom k percentuálnemu vyjadreniu počtu vstupných pixelov obrazu (osa  $x$ ).

### 6.1 Algoritmus OMP

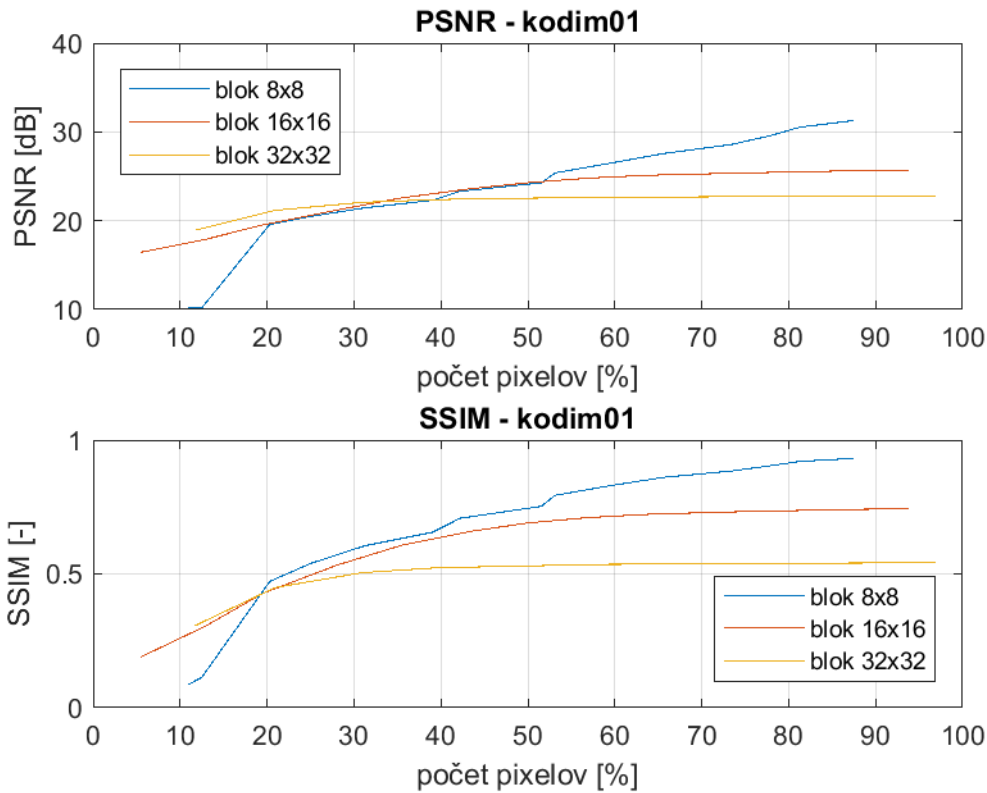
Táto kapitola sa zaoberá rekonštrukčnou kvalitou algoritmu OMP. Tým je myslené, že na vstupný obraz nie je aplikovaná žiadna maska. V simulácii na rekonštrukciu obrazu sú volené dva vstupné parametre. Jedným z nich je parameter  $K$ , ktorý určuje veľkosť jedného bloku, napr. 8x8, 16x16, atď. Druhým vstupným parametrom je parameter  $n$ , ktorým sa v tomto prípade nastavuje celkový počet vstupných dát, z ktorých následne prebieha rekonštrukcia vstupného obrazu. Parametrom  $n$  sa určuje počet pixelov v jednom bloku.

Dosiahnuté výsledné hodnoty pre dva obrazy (kodim01 a kodim23) sú zobrazené v grafoch 13 a 14. Na ose  $x$  je v percentách zobrazené množstvo vstupných dát pre následnú rekonštrukciu. Na ose  $y$  sú vynesené dosiahnuté hodnoty objektívnych metrík pre meranie kvality obrazu.

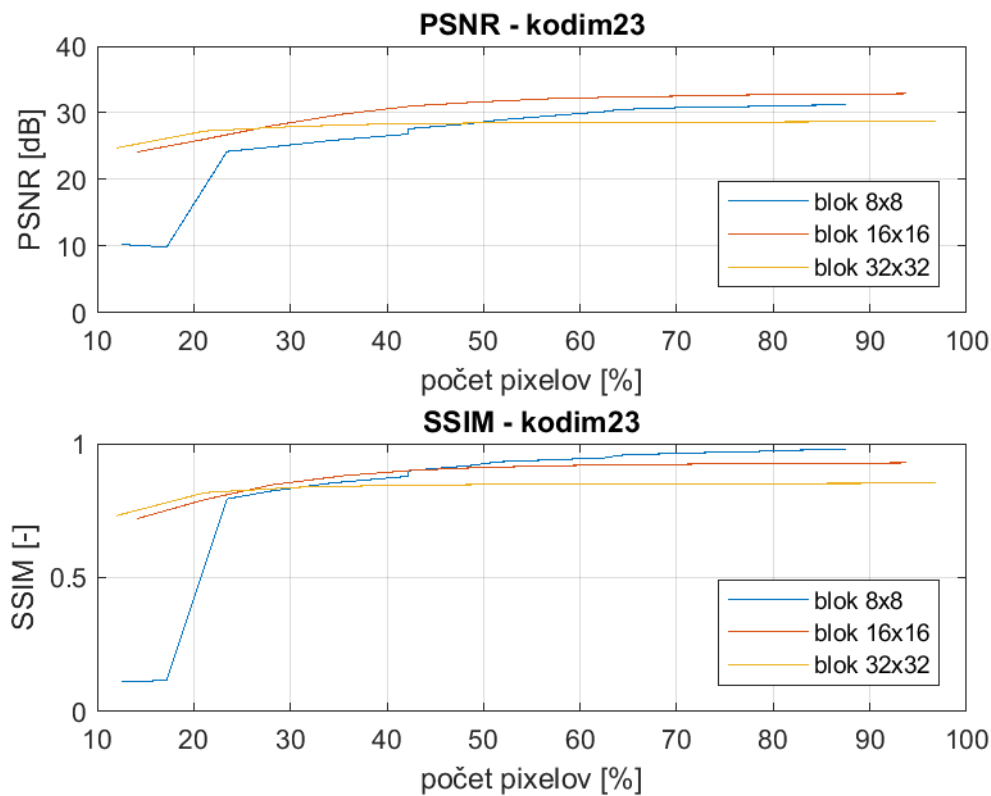
V oboch grafoch môžeme pozorovať, že pri voľbe malého bloku a pri malom počte vstupných dát dosahuje algoritmus slabé výsledky. Je to spôsobené nedostatočným počtom pixelov v jednom bloku. Okrem toho, pokiaľ sú ešte aj nevhodné rozmiestnené napríklad blízko vedľa seba, tak vznikne celý chybový blok, pretože algoritmus nemá dostatok vstupných dát na rekonštrukciu. Chybové bloky sú zobrazené na obr. 15.

---

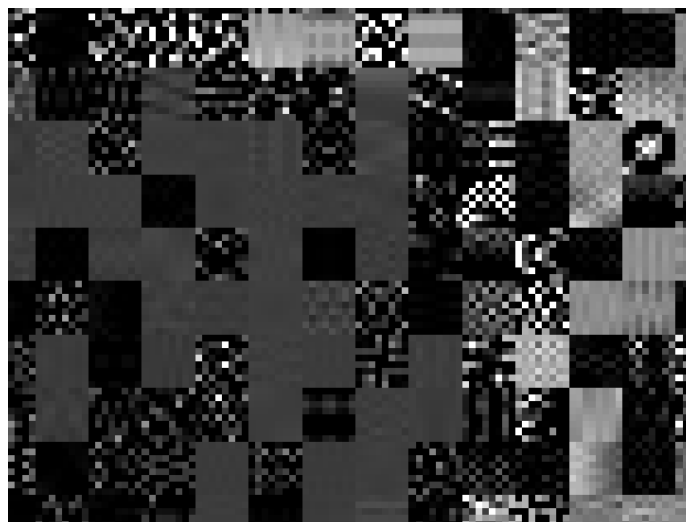
<sup>2</sup> Databáza je voľne dostupná na adrese: <http://r0k.us/graphics/kodak/>



Obrázok 14: Závislosť PSNR a SSIM na počte vstupných pixelov pri rekonštrukcii šedotónového obrazu kodim01

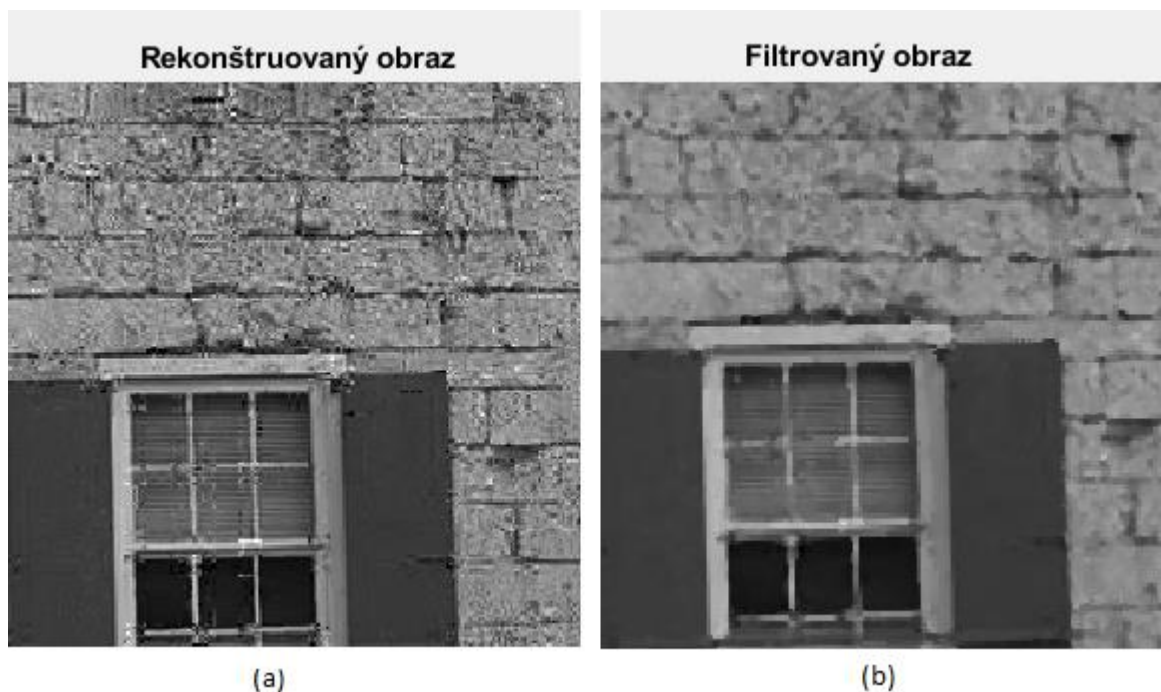


Obrázok 13: Závislosť PSNR a SSIM na počte vstupných pixelov pri rekonštrukcii šedotónového obrazu kodim23

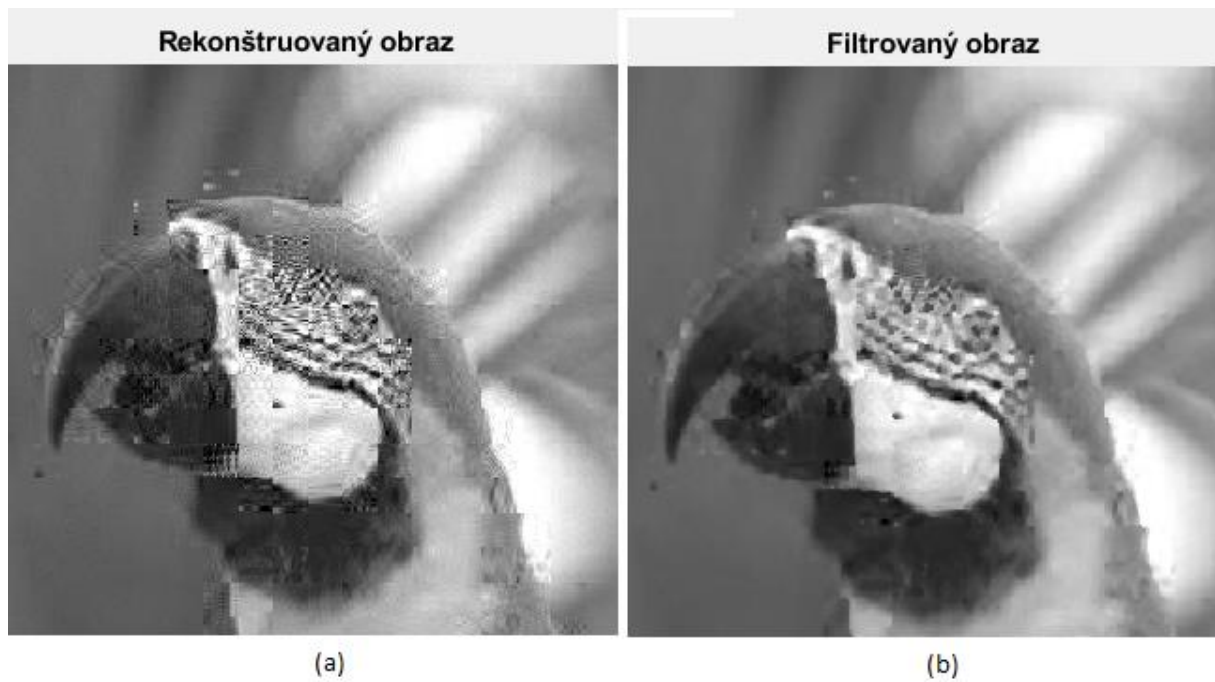


Obrázok 15: Ukážka chybových blokov pri simulácii s parametrami  $K=8$ ,  $n=8$ ;

Názorná ukážka detailu zrekonštruovaného obrazu je na obr. 16(a) a 17(a). Pri pohľade na obraz je možné pozorovať, že značným nedostatkom tejto rekonštrukčnej metódy je prechod medzi jednotlivými blokmi, keďže aj transformácia aj celý rekonštrukčný proces prebieha blok po bloku. Na zjemnenie prechodu medzi jednotlivými blokmi je možné následne po rekonštrukcii využiť niektorý z filtrov, na obr. 16(b) a 17(b) je zobrazené vyhladenie mediánovým filtrom, ktoré prináša celkom uspokojujivé výsledky pre náš vizuálny systém.



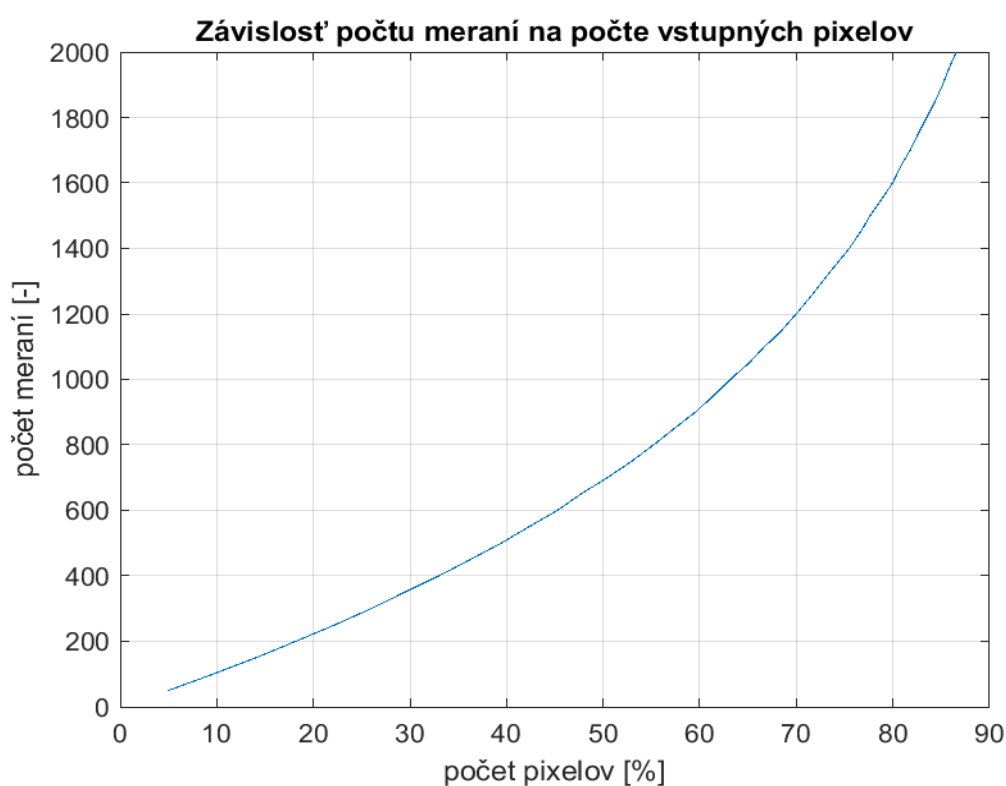
Obrázok 16: (a) Rekonštruovaný obraz s parametrami  $K=16$ ,  $n=100$ ; dosiahnuté hodnoty  $PSNR = 22,0651$  a  $SSIM = 0,6080$   
(b) Filtrovaný obraz mediánovým filtrom o veľkosti  $[3 \ 3]$ ; dosiahnuté hodnoty  $PSNR = 23,4856$  a  $SSIM = 0,6244$



Obrázok 17: (a) Rekonštruovaný obraz s parametrami  $K=16$ ,  $n=100$ ; dosiahnuté hodnoty  $PSNR = 29,7383$  a  $SSIM = 0,8724$   
(b) Filtrovaný obraz mediánovým filtrom o veľkosti  $[3 \ 3]$ ; dosiahnuté hodnoty  $PSNR = 31,5264$  a  $SSIM = 0,9114$

## 6.2 Simulácie jednopixelovej kamery

Model simulácie jednopixelovej kamery sa skladá z jednotlivých blokov, ktoré už boli popísané v kapitole 5.1. Simulácia vyžaduje nastavenie troch vstupných parametrov. Prvým z nich je parameter  $K$ , ktorý určuje tak isto ako v predchádzajúcom prípade veľkosť jedného bloku. Druhým vstupným parametrom je parameter  $n$ , ktorý v tomto prípade predstavuje počet meraní. Pri každom meraní sa náhodne vygeneruje nová vzorkovacia maska. Väčší počet meraní vygeneruje väčší počet vzorkovacích masiek a z väčšieho počtu vzorkovacích masiek dostaneme väčšie množstvo vstupných dát. Z dôvodu väčšej časovej náročnosti bol pri simulácii dbaný väčší dôraz na percentuálne zastúpenie vstupných dát, ako na adekvátny počet meraní. Na obr. 18 je zobrazená závislosť počtu meraní na percentuálnom zastúpení vstupných dát.

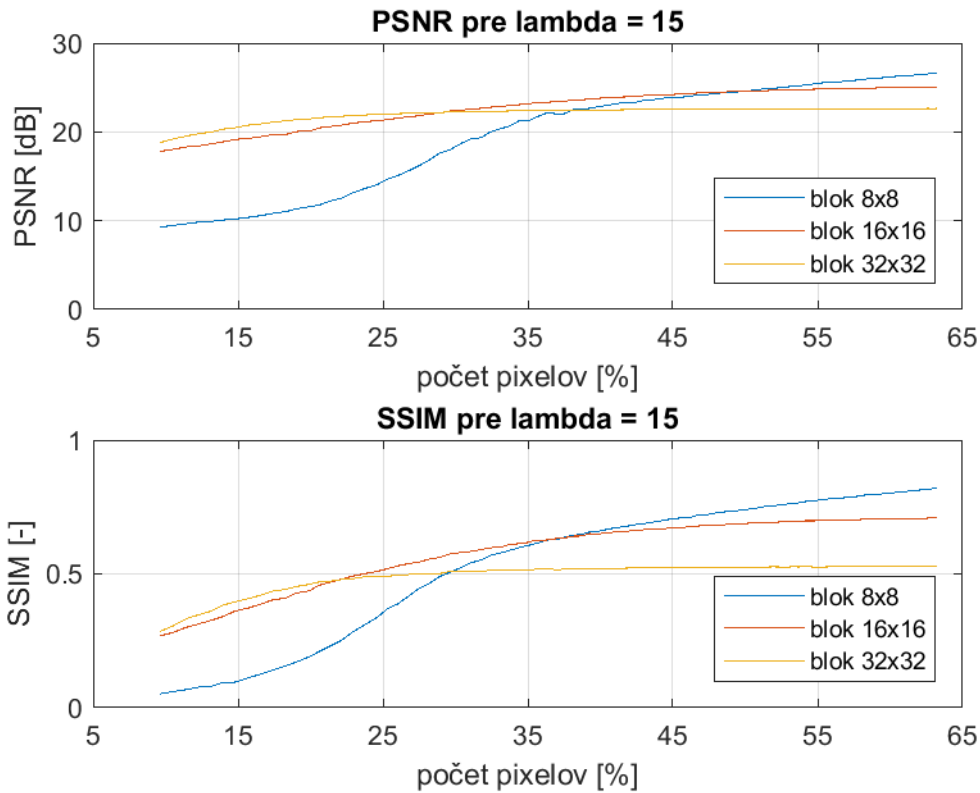


Obrázok 18: Závislosť počtu meraní na počte vstupných pixelov

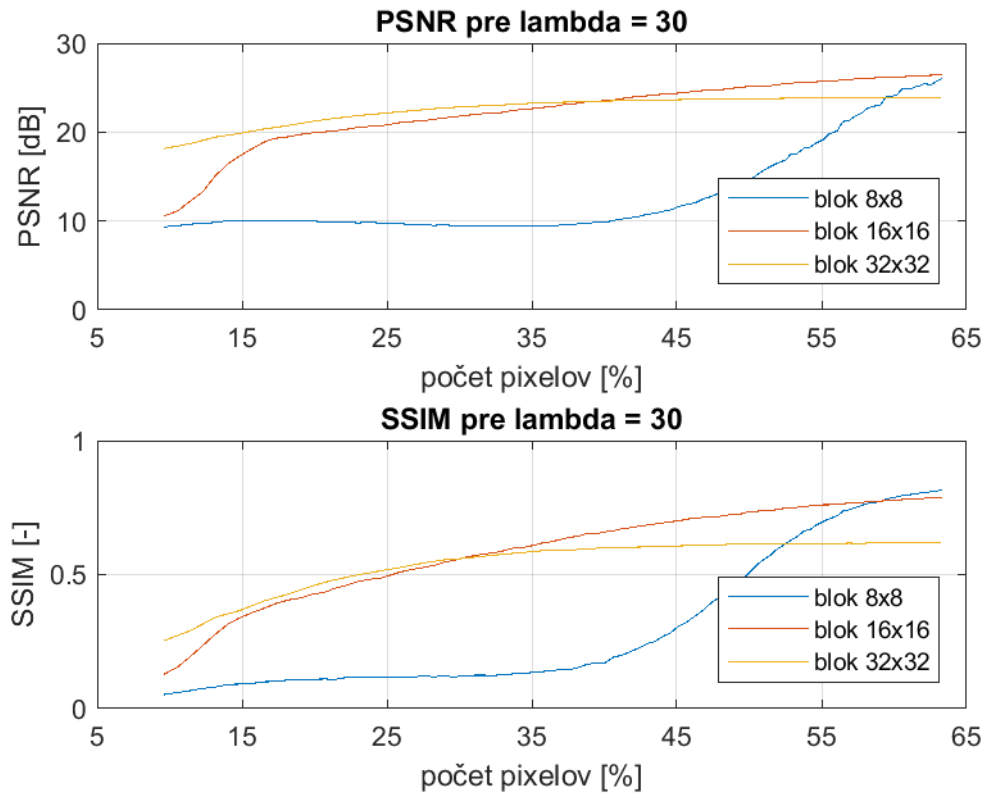
Posledným vstupným parametrom je parameter  $\lambda$  (lambda). Ten udáva maximálny počet iterácií v rekonštrukčnom algoritme OMP. Hodnota parametru  $\lambda$  výrazne ovplyvňuje čas trvania celej simulácie. Názorná ukážka závislosti parametru  $\lambda$  na celkovom čase simulácie je zobrazená v tab. 2.

-	$\lambda$ [-]	15	30	50
ČB	t[s]	34,13	84,85	271,85
RGB	t[s]	96,47	256,58	815,05

Tabuľka 2: Časová závislosť na parametri  $\lambda$ , pri veľkosti bloku  $K=16$ , počte meraní  $n=1000$  a pri rekonštrukcii šedotónového obrazu kodim23

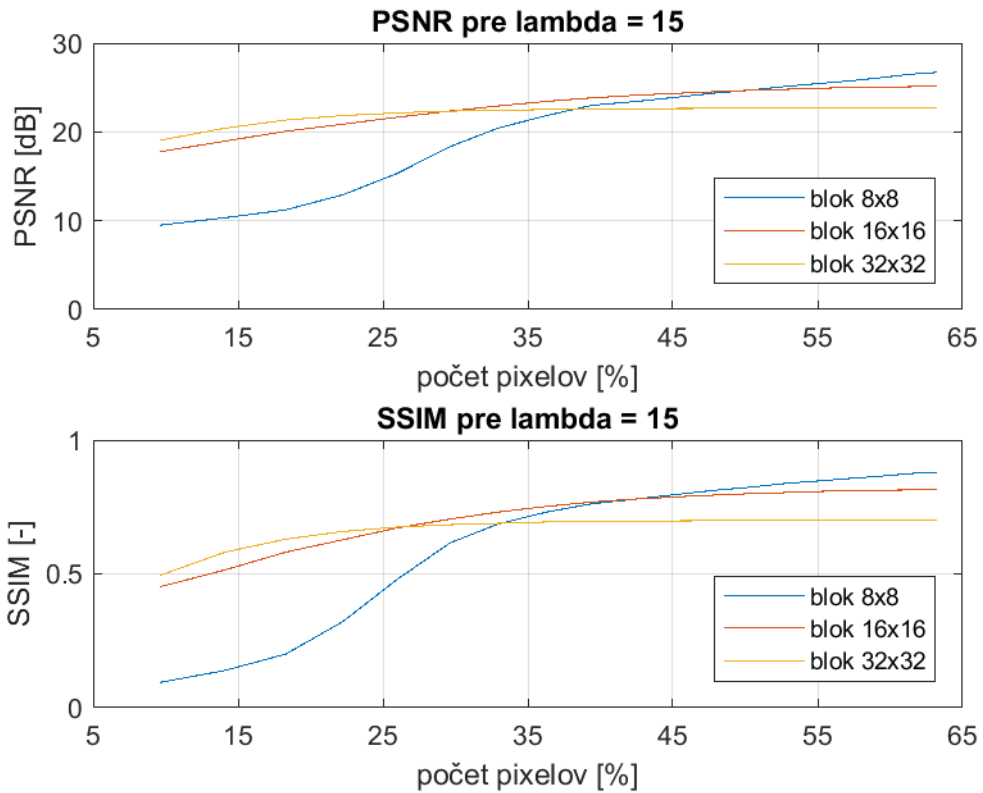


Obrázok 19: Závislosť PSNR a SSIM vzhľadom k počtu vstupných pixelov pre parameter  $\lambda = 15$ , pri rekonštrukcii šedotónového obrazu kodim01

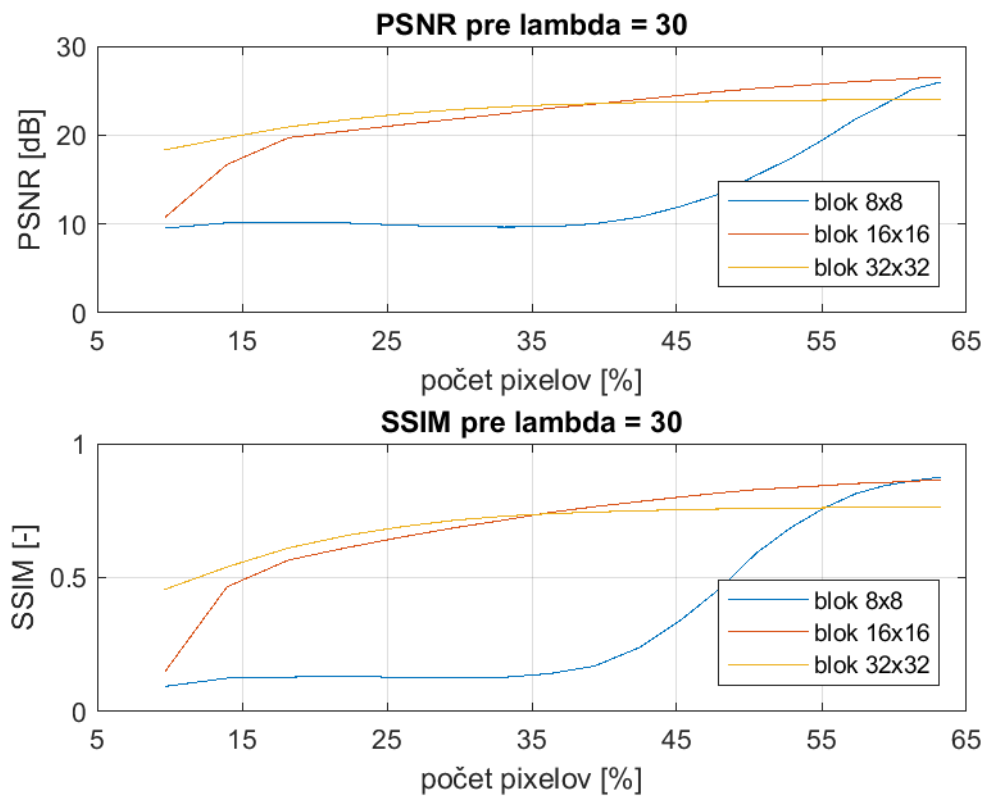


Obrázok 20: Závislosť PSNR a SSIM vzhľadom k počtu vstupných pixelov pre parameter  $\lambda = 30$ , pri rekonštrukcii šedotónového obrazu kodim01





Obrázok 21: Závislosť PSNR a SSIM vzhľadom k počtu vstupných pixelov pre parameter  $\lambda = 15$ , pri rekonštrukcii farebného obrazu kodim01



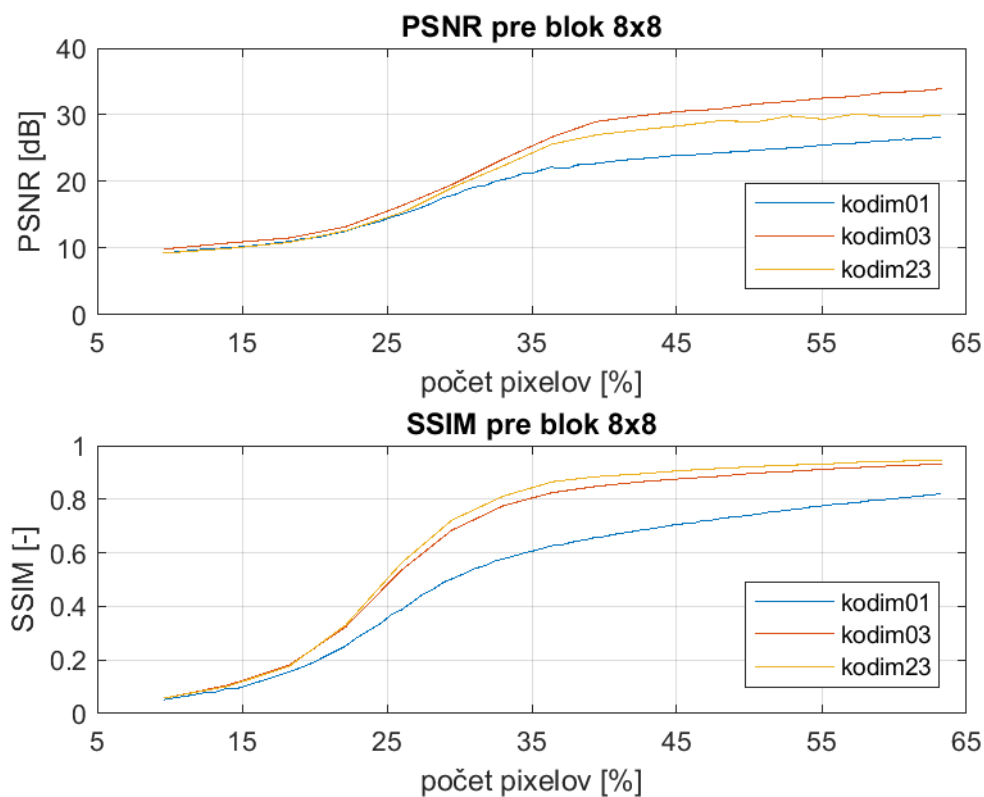
Obrázok 22: Závislosť PSNR a SSIM vzhľadom k vstupnému počtu pixelov pre parameter  $\lambda = 30$ , pri rekonštrukcii farebného obrazu kodim01

Na obr. 19 až 22 je možné pozorovať dosiahnuté výsledky pre dve rôzne hodnoty parametra  $\lambda$ . Na simuláciu bol ako vstupný obraz použitý obraz kodim01 z databázy Kodaku. Obr. 19 a 20 obsahuje výsledné hodnoty pre šedotónovú rekonštrukciu obrazu a obr. 21 a 22 predstavuje dosiahnuté výsledné hodnoty pre farebnú rekonštrukciu obrazu.

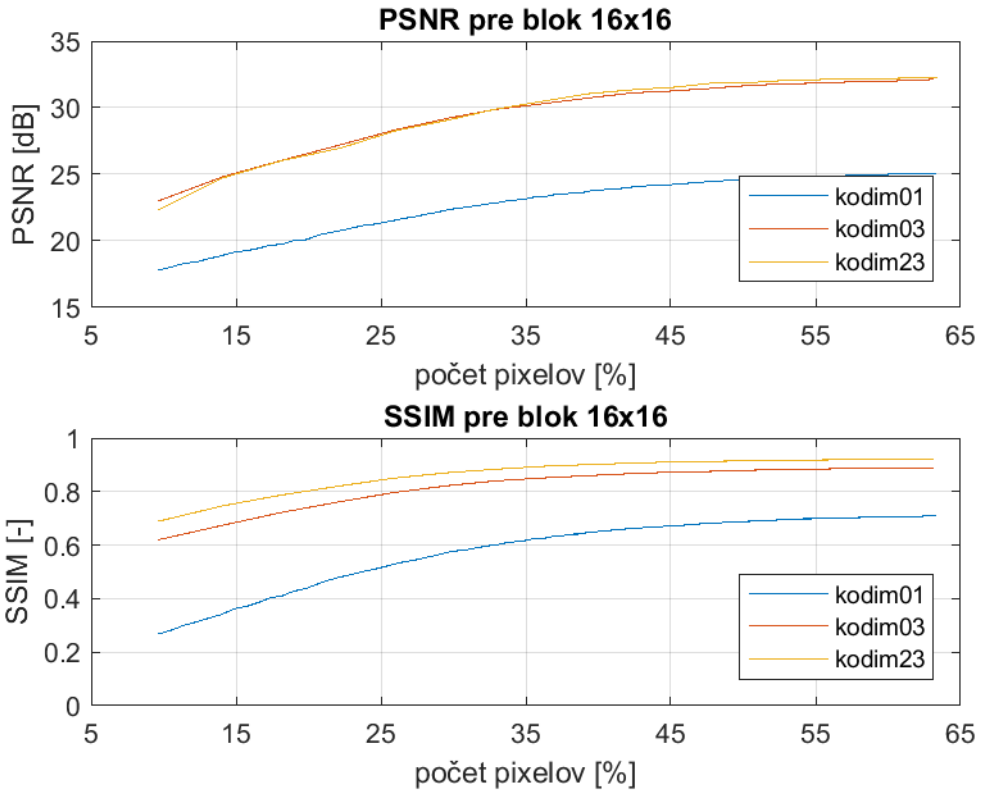
Z grafov je možné pozorovať, že pre závislosť PSNR na počte vstupných pixelov nezáleží na tom, či sa rekonštruje farebný alebo šedotónový obraz. Dosiahnuté hodnoty PSNR sú rovnaké. Podobne je tomu aj pri závislosti parametru SSIM. Aj keď dosiahnuté hodnoty pre farebné prevedenie sú o niečo lepšie, tento malý rozdiel je zanedbateľný. To bude z toho dôvodu, že pri farebnom obraze algoritmus OMP vytvára rekonštrukciu pre každú farebnú zložku zvlášť, avšak na rovnakom princípe ako pre šedotónový obraz.

Pri zmene parametra  $\lambda$  je možné pozorovať rozdiely vo výsledných hodnotách. Najzásadnejšie sú rozdiely pri voľbe menšieho bloku 8x8, kde algoritmus pri nastavení parametra  $\lambda = 30$ , obsahuje veľké množstvo chybových blokov aj pri väčšom počte vstupných dát v porovnaní s druhým nastavením. Avšak pri obsiahlejšom počte vstupných dát (nad 55%) dosahuje o niečo lepšie výsledné kvality. Taktiež je potrebné pripomenúť fakt, na základe tab. 2, že jedna simulácia pre hodnotu  $\lambda = 30$  trvá približne 2,5 krát dlhšie ako pri hodnote  $\lambda = 15$ .

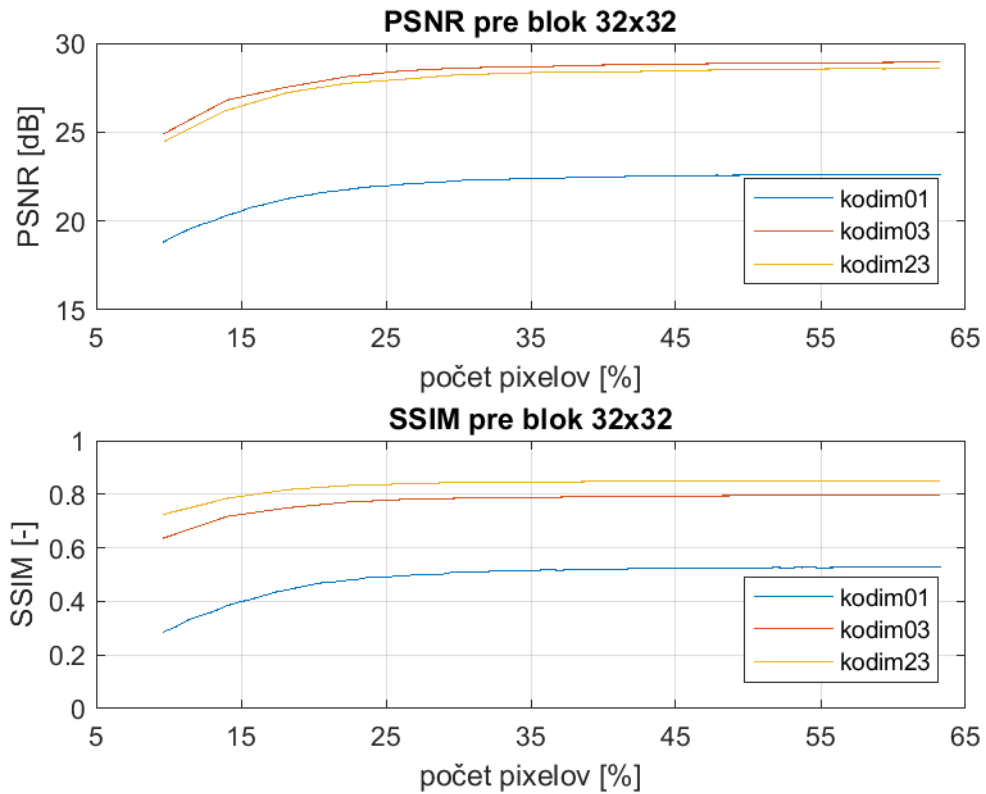
V nasledujúcich grafoch je možné pozorovať porovnania dosiahnutých hodnôt objektívnych metrick pre rôzne vstupné obrazy.



Obrázok 23: Porovnanie závislosti PSNR a SSIM vzhľadom k počtu vstupných pixelov na rôzne vstupné obrazy pre parametre  $K=8$  a  $\lambda = 15$



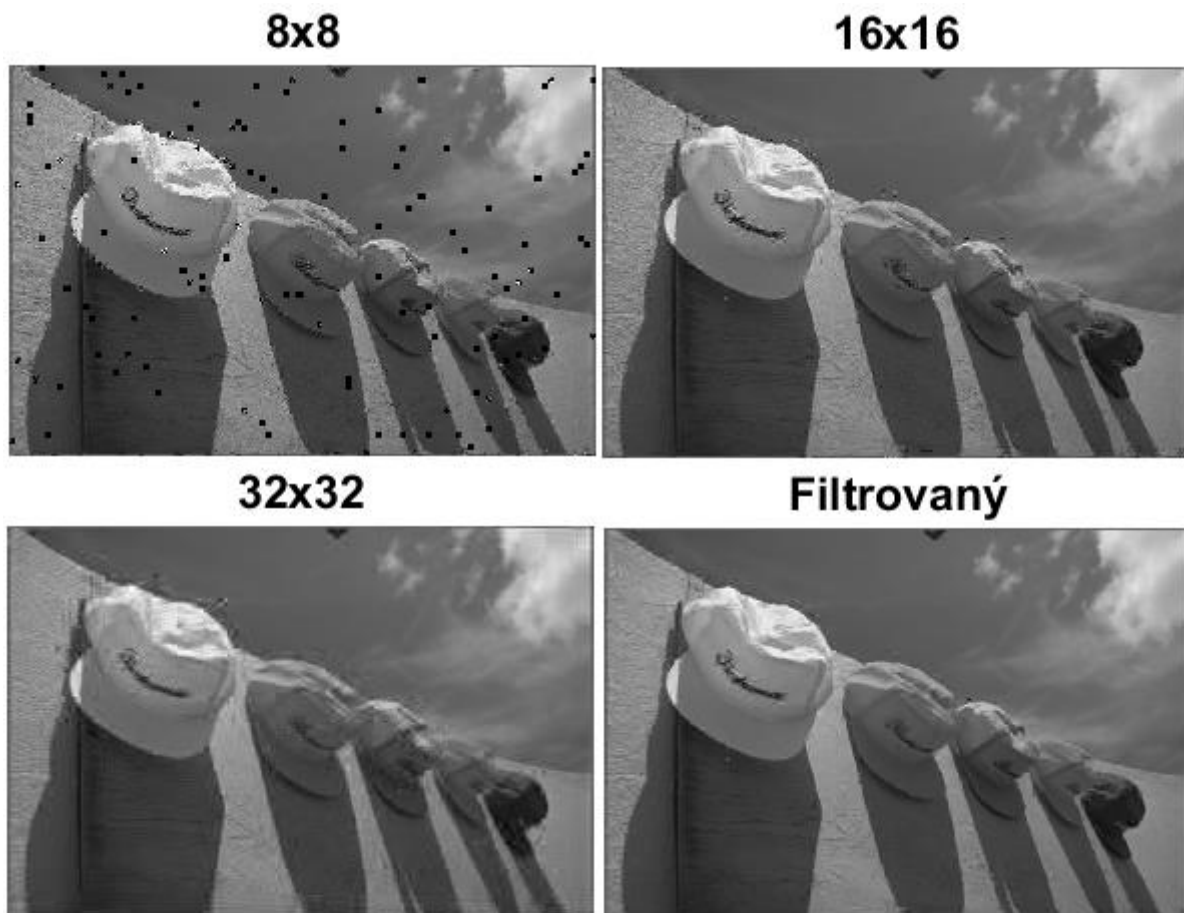
Obrázok 24: Porovnanie závislosti PSNR a SSIM vzhľadom k počtu vstupných pixelov na rôzne vstupné obrazy pre parametre  $K = 16$  a  $\lambda = 15$



Obrázok 25: Porovnanie závislosti PSNR a SSIM vzhľadom k počtu vstupných pixelov na rôzne vstupné obrazy pre parametre  $K = 32$  a  $\lambda = 15$

Na základe obr. 23 až 25 je možné povedať, že algoritmus je jednoznačne obsahovo závislý na vstupnom obraze. Ako bolo vyššie v texte už niekoľko krát uvedené, celý proces komprimovaného snímania závisí okrem vhodného nastavenia jednotlivých parametrov pri rekonštrukcii, taktiež aj na vlastnostiach vstupného obrazu. Obraz by mal byť ľahko komprimovateľný, tým je myslené, že by mal byť riedky v niektorých oblastiach transformácie.

Nasleduje názorná ukážka rekonštruovaných obrazov zo simulácie pre rôzne nastavenia veľkosti bloku.



Obrázok 26: Porovnanie pri rovnakom počte meraní  $n=400$  tj. 33,1% vstupných dát a rôznych veľkostiach blokov; na filtráciu bol použitý mediánový filter o veľkosti [3 3] na obraz s veľkosťou bloku 16x16

Dosiahnuté hodnoty objektívnych metrík:

Názov	PSNR [dB]	SSIM [-]
8x8	23,8701	0,7789
16x16	29,7309	0,8374
32x32	28,691	0,788
Filtrovaný	30,5077	0,8513

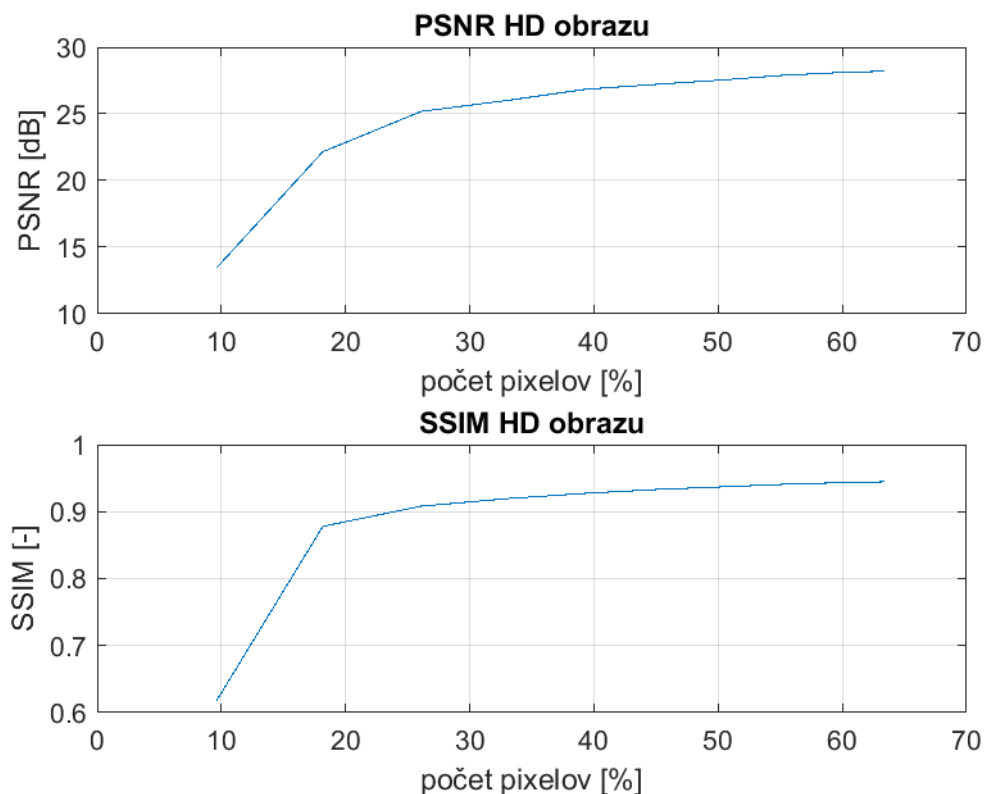
Tabuľka 3: Prehľad dosiahnutých hodnôt pre objektívne metríky

Z obr. 26 je možné pozorovať, že najlepšie výsledky bez filtrácie dosahuje obraz s veľkosťou bloku 16x16. Pri voľbe bloku 8x8 vznikajú miestami chybové bloky a pri voľbe bloku 32x32 sú viac rozmazané (nesprávne zrekonštruované) prechody pri objektoch.

### 6.3 Rekonštrukcia HD obrazu

Model jednopixelovej kamery bol taktiež simulovaný aj so vstupným obrazom vyššej kvality. Na simuláciu bol použitý obraz s rozlíšením 1920x1200 pixelov. Pri vstupnom obraze s takto vysokým rozlíšením nastáva problém pri aplikácii vzorkovacej masky. Vo všeobecnosti sa uvádza, že zariadenie DMD obsahuje niekoľko stoviek tisíc mikrozrkadiel. V literatúra [5] je špecifikované toto číslo na 700 000. Pri štandardnom rozlíšení obrazov z databáze Kodak reprezentovalo jedno zrkadlo jeden pixel vstupného obrazu. V tomto prípade to však nie je možné, pretože spomínaný vstupný obraz vo vyššej kvalite obsahuje viac ako 2 300 000 pixelov, čo zďaleka prekračuje počet zrkadiel. Preto bolo nutné pozmeniť aplikovanie vzorkovacej masky. V simulácii pre obraz s rozlíšením 1920x1200 pixelov predstavuje jedno zrkadlo plochu o rozlohe 2x2 pixelov. Tým bol vyriešený problém s nedostatočným počtom zrkadiel.

Z dôvodu výpočtovej náročnosti nebola simulácia vyhotovená pre rôzne nastavenia parametrov. Ukážka výsledných hodnôt rekonštrukcie je zobrazená v grafe 27. Simulácia bola prevedená o veľkosti bloku  $K = 16$  a s hodnotou  $\lambda = 15$ . Na obr. 28 je zobrazený výsledok rekonštrukcie. Pri simulácii bol nastavený počet meraní na  $n = 300$ , čo pri vstupnom HD obraze odpovedá 26% vstupných dát. Pri podrobnom skúmaní je možné spozorovať nedostatky, ktoré sú zobrazené na obr. 29(a) a 30(a). Podobne ako v kapitole 6.1 sú tieto nedostatky spôsobené prechodmi medzi blokmi. Tie môžeme potlačiť takisto napríklad aplikáciou mediánového filtra.



Obrázok 27: : Závislosť PSNR a SSIM vzhľadom k počtu pixelov pri rekonštrukcii farebného HD obrazu



Obrázok 29: Zrekonštruovaný obraz, pri simulácii bolo použitých 26% vstupných dát



(a)



(b)

Obrázok 30: (a) zrekonštruovaný obraz (b) filtrovaný obraz



(a)

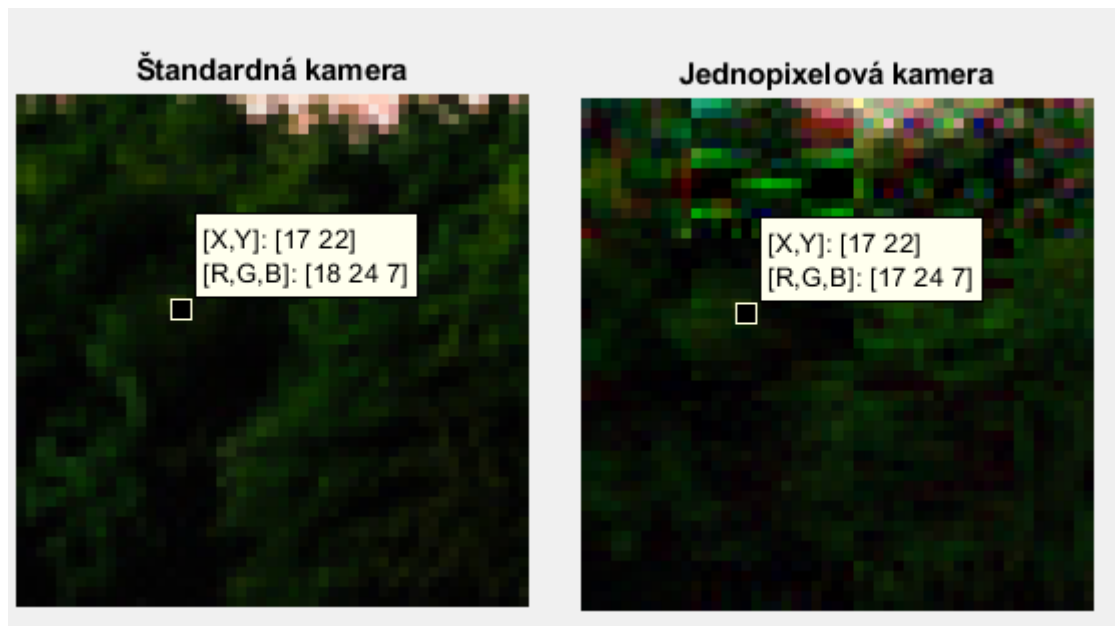


(b)

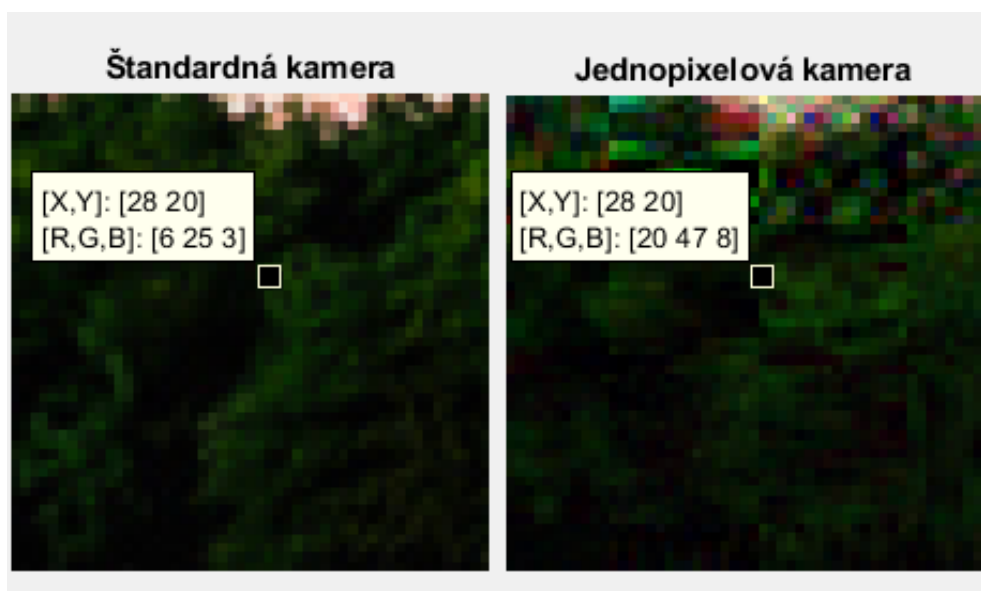
Obrázok 28: (a) zrekonštruovaný obraz (b) filtrovaný obraz

## 6.4 Porovnanie jednopixelovej kamery so štandardnou kamerou

Taktiež je možné porovnať model jednopixelovej kamery so štandardnou kamerou. Pri štandardnej kamere by sme uvažovali použitie rovnákeho objektívu ako pri jednopixelovej kamere, takže objektív, ktorý má gaussovskú rozptylovú funkciu (PSF). Porovnanie prebehlo pri simulácii jednopixelovej kamery s nastavením bloku o veľkosti  $K = 16$ , s hodnotou  $\lambda = 15$ . Pri simulácii bol nastavený počet meraní na  $n = 300$ , čo pri vstupnom HD obraze odpovedá 26% vstupných dát.



Obrázok 31: Porovnanie štandardnej a jednopixelovej kamery, poloha bodu [17 22] sa nenachádza na okraji bloku a pôvodná hodnota na vstupnom obraze je [19 21 10]



Obrázok 32: Porovnanie štandardnej a jednopixelovej kamery, poloha bodu [28 20] sa nachádza na okraji bloku a pôvodná hodnota na vstupnom obraze je [5 25 0]

Na obr. 31 a 32 môžeme pozorovať dosiahnuté hodnoty pixelov v jednotlivých farebných priestoroch. Tak ako v predchádzajúcich prípadoch, pri jednopixelovej kamere sa vyskytujú prechody medzi jednotlivými blokmi, ktoré výrazne ovplyvňujú dosiahnuté hodnoty. Preto záleží, či sa pri porovnaní vyberie pixel, ktorý je na prechode blokov (viz obr. 29), alebo viac v jeho vnútri (viz obr. 28). Pokiaľ porovnávame pixely, ktoré sú na prechode, tak tie dosahujú odlišné hodnoty, zatiaľ čo pixely vo vnútri bloku dosahujú takmer totožné. V tab. 3 je možné pozorovať dosiahnuté hodnoty metrík pri rôznych porovnaníach.

	rec vs orig	rec vs stand	stand vs orig
PSNR [dB]	25,2767	26,2142	38,0729
SSIM [-]	0,9084	0,9196	0,9939

*Tabuľka 4: Porovnanie dosiahnutých hodnôt metrík pre rôzne porovnaníach;  
rec = rekonštruovaný obraz, orig = originálny obraz, stand= obraz zo štandardnej kamery*



## Záver

Táto diplomová práca sa zaoberá tematikou komprimovaného snímania. Cieľom tejto práce bolo podať prehľad metód komprimovaného snímania a ich využitie pri snímaní a spracovaní obrazu.

V teoretickej časti je podaný úvod do problematiky riedkej reprezentácie signálov, kde sú zhrnuté základné definície, ktoré sú nevyhnutné pre určenie vhodného a jedinečného  $k$ -riedkeho riešenia systému. Riedku reprezentáciu signálu chápeme ako systém nedourčených lineárnych rovníc, ktorý sa snažíme za určitých podmienok a predpokladov vyriešiť. Systém nedourčených lineárnych rovníc tvorí sústava rovníc, ktorých počet je menší ako počet neznámych. V množine riešení týchto rovníc hľadáme riešenie, ktoré má čo najviac nulových hodnôt, teda tzv. riedke riešenie. Na nájdenie riedkeho riešenia existuje množstvo metód. Niekoľko základných metód bolo uvedených a následne stručne popísaných v tejto práci.

Ako bolo v práci uvedené, z teórie komprimovaného snímania vyplýva, že pokiaľ je signál riedky a komprimovateľný v niektorých oblastiach transformácie, tak môže byť rekonštruovaný z menšieho počtu vzoriek signálu. Komprimované snímanie využíva iný prístup pri snímaní obrazu ako sa štandardne používa. V prístupe komprimovaného snímania sa signál sníma lineárne a neadaptívne. Dôsledkom toho, signál postačí nasnímať iba toľko krát, koľko je potrebné k jeho rekonštrukcii.

Komprimované snímanie si postupne nachádza uplatnenie pri snímaní a spracovaní obrazu. Táto práca bola zameraná na využitie komprimovaného snímania pri jednopixelovej kamere. Cieľom praktickej časti práce bolo navrhnúť a implementovať simuláciu modelu jednopixelovej kamery. Navrhnutý model sa podarilo implementovať a úspešne rekonštruovať vstupné obrazy. Dosiahnuté výsledky ako aj názorné ukážky zrekonštruovaných obrazov sa nachádzajú v šiestej kapitole.

Tento navrhnutý model je možné rozdeliť do dvoch základných častí, časti vzorkovania (spracovania) a časti rekonštrukcie. V prvej časti je snímaný vstupný obraz, na ktorý sa následne aplikuje vzorkovacia maska. Tá predstavuje aplikáciu DMD modelu, ktorý sa skladá z veľkého množstva malinkých zrkadiel. Podľa nastavenia zrkadiel sa svetlo smeruje buď smerom k detektoru alebo mimo neho.

V druhej časti to je proces rekonštrukcie, kde je najskôr prevedená bloková transformácia. Pretože základom celej teórie komprimovaného snímania, je predpoklad, že daný obraz je komprimovateľný resp. riedky v niektorých oblastiach transformácie. V simulácii je použitá diskretná kosínová transformácia (DCT). Následne je použitý jeden zo základných rekonštrukčných algoritmov, algoritmus Orthogonal Matching Pursuit (OMP). Jeho teoretický rozbor ako aj blokové schéma je popísané vyššie v texte, konkrétne v kapitole 3.2.3. Tento algoritmus bol vybraný pre jeho dobrý pomer medzi výpočtovou náročnosťou a výslednou kvalitou rekonštruovaného obrazu.

Zrekonštruované obrazy dosahujú na prvý pohľad celkom uspokojivé výsledky. Avšak pri podrobnejšom skúmaní je možné pozorovať dva typy nedostatkov. Prvým sú chybové bloky. Tie sú spôsobené nedostatočným počtom vstupných pixelov v jednom bloku alebo ich nevhodným rozmiestnením. Nevhodným rozmiestnením sú myslené zhluky vstupných pixelov. Druhým nedostatkom je prechod medzi jednotlivými blokmi, keďže aj transformácia aj rekonštrukčný algoritmus prebieha blok po bloku. Na zjemnenie prechodu medzi jednotlivými blokmi je možné následne po rekonštrukcii použiť filtráciu. V tejto práci bol na vyhladenie prechodu medzi blokmi použitý mediánový filter. Po jeho aplikácii na zrekonštruovaný obraz prechody medzi jednotlivými blokmi nepôsobia až tak rušivo pre náš vizuálny systém.

## Zdroje

- [1] DONOHO, David L.; ELAD, Michael. Optimally sparse representation in general (nonorthogonal) dictionaries via  $\ell_1$  minimization. *Proceedings of the National Academy of Sciences*, 2003, 100.5: 2197-2202.
- [2] BRUCKSTEIN, Alfred M.; DONOHO, David L.; ELAD, Michael. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM review*, 2009, 51.1: 34-81.
- [3] CANDÈS, Emmanuel J.; WAKIN, Michael B. An introduction to compressive sampling. *IEEE signal processing magazine*, 2008, 25.2: 21-30.
- [4] ELDAR, Yonina C.; KUTYNIOK, Gitta (ed.). *Compressed sensing: theory and applications*. Cambridge University Press, 2012.
- [5] FORNASIER, Massimo; RAUHUT, Holger. *Handbook of Mathematical Methods in Imaging*. Chapter 6 Compressive Sensing, 2011, 6: 187-228.
- [6] HRBÁČEK, Radek, et al. Řídké reprezentace signálů: úvod do problematiky. *Elektrorevue-Internetový časopis*, 2011, s. 1-10.
- [7] HRBÁČEK, Radek, et al. Řídké reprezentace signálů: komprimované snímání. *Elektrorevue-Internetový časopis*, 2011.
- [8] CHEN, Scott Shaobing; DONOHO, David L.; SAUNDERS, Michael A. Atomic decomposition by basis pursuit. *SIAM review*, 2001, 43.1: 129-159.
- [9] KARAHANOGU, Nazim Burak; ERDOGAN, Hakan. A\* orthogonal matching pursuit: Best-first search for compressed sensing signal recovery. *Digital Signal Processing*, 2012, 22.4: 555-568.
- [10] TROPP, Joel A.; GILBERT, Anna C. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 2007, 53.12: 4655-4666.
- [11] GORODNITSKY, Irina F.; RAO, Bhaskar D. Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm. *IEEE Transactions on signal processing*, 1997, 45.3: 600-616.
- [12] HRBÁČEK, Radek. *Využití řídké reprezentace signálu při snímání a rekonstrukci v nukleární magnetické rezonanci*. Brno, 2013. Diplomová práce
- [13] M. Elad: *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, 1st ed. Berlin, Germany: SpringerVerlag, 2010.
- [14] DUARTE, Marco F., et al. Single-pixel imaging via compressive sampling. *IEEE signal processing magazine*, 2008, 25.2: 83-91.
- [15] ELAD, Michael; AHARON, Michal. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 2006, 15.12: 3736-3745.

- [16] THOMPSON, Andrew. Compressive single-pixel imaging. In: Proceedings of the 2nd IMA Conference on Mathematics in Defence, Defence Academy, Shrivenham, UK. 2011.
- [17] DONOHO, David L., et al. Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 2012, 58.2: 1094-1121.
- [18] BLUMENSATH, Thomas; DAVIES, Mike E. Stagewise weak gradient pursuits. *IEEE Transactions on Signal Processing*, 2009, 57.11: 4333-4346.
- [19] NEEDELL, Deanna; VERSHYNIN, Roman. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. *IEEE Journal of selected topics in signal processing*, 2010, 4.2: 310-316.
- [20] BLUMENSATH, Thomas; DAVIES, Mike E. Normalized iterative hard thresholding: Guaranteed stability and performance. *IEEE Journal of selected topics in signal processing*, 2010, 4.2: 298-309.
- [21] BLUMENSATH, Thomas; DAVIES, Mike E. Gradient pursuits. *IEEE Transactions on Signal Processing*, 2008, 56.6: 2370-2382.
- [222] BEGON, Martin; CIAPALA, Richard; DEAKI, Zoltan. Reliability of MEMS: Case study January 30th 2007- Presentation transcript at <http://slideplayer.com/slide/7368696/>
- [23] TAKHAR, Dharmpal, et al. A new compressive imaging camera architecture using optical-domain compression. In: *Computational Imaging IV*. International Society for Optics and Photonics, 2006. p. 606509.
- [24] AHARON, Michal; ELAD, Michael; BRUCKSTEIN, Alfred. The K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 2006, 54.11: 4311-4322.

## Zoznam použitých skratiek

BP	Basis Pursuit
LARS	Least Angle Regression
IRLS	Iterative Reweighted Least Squares
FOCUSS	FOCaI Underdetermined System Solver
MP	Matching Pursuit
OMP	Orthogonal Matching Pursuit
GP	Gradient Pursuit
CGP	Conjugate Gradient Pursuit
StOMP	Stagewise Orthogonal Matching Pursuit
StWGP	Stagewise Weak Gradient Pursuit
ROMP	Regularized OMP
ORMP	Order Recursive Matching Pursuit
IHT	Iterative Hard Thresholding
NIHT	Normalized Iterative Hard Thresholding
CoSaMP	Compressive Sampling Matching Pursuit
SP	Subspace Pursuit
A*OMP	A* Orthogonal Matching Pursuit
SLM	Spatial Light Modulator
DMD	Digital Micromirror Device
CS	Compressive Sensing
CCD	Charge-coupled device
CMOS	Complementary metal–oxide–semiconductor
A/D	Analogovo Digitálny prevodník
PSF	Point spread function
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity
ČB	Čierno-biely
HD	High-definition

## Zoznam obrázkov

Obrázok 1: [6] Schéma nedourčeného systému rovníc .....	11
Obrázok 2: [7] Ilustračné schéma komprimovaného snímania .....	15
Obrázok 3: Prehľad rekonštrukčných metód komprimovaného snímania .....	17
Obrázok 4: [4] Rámcové schéma všeobecného postupu hladných algoritmov .....	21
Obrázok 5: [4] Rámcové schéma algoritmu Matching Pursuit (MP) .....	22
Obrázok 6: [4] Rámcové schéma algoritmu Orthogonal Matching Pursuit (OMP).....	23
Obrázok 7: [4] Rámcové schéma algoritmu Directional/Gradient Pursuit .....	24
Obrázok 8: [4] Rámcové schéma algoritmu Iterative Hard Thresholding.....	26
Obrázok 9: [16] Schéma jednopixelovej CS kamery .....	29
Obrázok 10: [22] Schéma dvoch zrkadiel DMD zariadenia.....	30
Obrázok 11: [14] Príklad rozmiestnenia hardwaru jednopixelovej kamery .....	32
Obrázok 12: Schéma simulácie jednopixelovej kamery .....	33
Obrázok 14: Závislosť PSNR a SSIM na počte vstupných pixelov pri rekonštrukcii šedotónového obrazu kodim23.....	36
Obrázok 13: Závislosť PSNR a SSIM na počte vstupných pixelov pri rekonštrukcii šedotónového obrazu kodim01.....	36
Obrázok 15: Ukážka chybových blokov pri simulácii s parametrami $K=8, n=8$ ;.....	37
Obrázok 16: (a) Rekonštruovaný obraz s parametrami $K=16, n=100$ ; dosiahnuté hodnoty PSNR =22,0651 a SSIM = 0,6080 .....	37
Obrázok 17: (a) Rekonštruovaný obraz s parametrami $K=16, n=100$ ; dosiahnuté hodnoty PSNR =29,7383 a SSIM = 0,8724 .....	38
Obrázok 18: Závislosť počtu meraní na počte vstupných pixelov.....	39
Obrázok 19: Závislosť PSNR a SSIM vzhľadom k počtu vstupných pixelov pre parameter $\lambda = 15$ , pri rekonštrukcii šedotónového obrazu kodim01 .....	40
Obrázok 20: Závislosť PSNR a SSIM vzhľadom k počtu vstupných pixelov pre parameter $\lambda = 30$ , pri rekonštrukcii šedotónového obrazu kodim01 .....	40
Obrázok 21: Závislosť PSNR a SSIM vzhľadom k počtu vstupných pixelov pre parameter $\lambda = 15$ , pri rekonštrukcii farebného obrazu kodim01.....	41
Obrázok 22: Závislosť PSNR a SSIM vzhľadom k vstupných počtu pixelov pre parameter $\lambda = 30$ , pri rekonštrukcii farebného obrazu kodim01.....	41
Obrázok 23: Porovnanie závislosti PSNR a SSIM vzhľadom k počtu vstupných pixelov na rôzne vstupné obrazy pre parametre $K=8$ a $\lambda = 15$ .....	42
Obrázok 24: Porovnanie závislosti PSNR a SSIM vzhľadom k počtu vstupných pixelov na rôzne vstupné obrazy pre parametre $K = 16$ a $\lambda = 15$ .....	43
Obrázok 25: Porovnanie závislosti PSNR a SSIM vzhľadom k počtu vstupných pixelov na rôzne vstupné obrazy pre parametre $K = 32$ a $\lambda = 15$ .....	43
Obrázok 26: Porovnanie pri rovnakom počte meraní $n=400$ tj.33,1% vstupných dát a rôznych veľkostiach blokov; na filtráciu bol použitý mediánový filter o veľkosti $[3 \ 3]$ na obraz s veľkosťou bloku $16 \times 16$ .....	44
Obrázok 27: : Závislosť PSNR a SSIM vzhľadom k počtu pixelov pri rekonštrukcii farebného HD obrazu .....	45
Obrázok 28: Zrekonštruovaný obraz, pri simulácii bolo použitých 26% vstupných dát.....	46
Obrázok 29: (a) zrekonštruovaný obraz (b) filtrovaný obraz .....	46

Obrázok 30: (a) zrekonštruovaný obraz (b) filtrovaný obraz .....	46
Obrázok 31: Porovnanie štandardnej a jednopixelovej kamery, poloha bodu [17 22] sa nenachádza na okraji bloku a pôvodná hodnota na vstupnom obraze je [19 21 10].....	47
Obrázok 32: Porovnanie štandardnej a jednopixelovej kamery, poloha bodu [28 20] sa nachádza na okraji bloku a pôvodná hodnota na vstupnom obraze je [5 25 0].....	47

## Zoznam tabuliek

Tabuľka 1: [4] Prehľad výpočtovej a pamäťovej náročnosti pre jednu iteráciu .....	25
Tabuľka 2: Časová závislosť na parametri $\lambda$ , pri veľkosti bloku $K=16$ , počte meraní $n=1000$ a pri rekonštrukcii šedotónového obrazu kodim23 .....	39
Tabuľka 3: Prehľad dosiahnutých hodnôt pre objektívne metriky .....	44
Tabuľka 4: Porovnanie dosiahnutých hodnôt metrík pre rôzne porovnania .....	48

## Zoznam príloh

popis_suborov.txt	textový dokument s popisom súborov
data.zip	obsahuje všetky namerané hodnoty (.mat) + skripty na vykreslenie grafov
rec_obr.zip	obsahuje ukážky zrekonštruovaných obrazov

### Matlab skripty:

cross.m	skript na počítanie lambda
mask.m	skript na vytvorenie čiernobielej masky pre klasické rozlíšenie obrazov
mask_gray_hd.m	skript na vytvorenie čiernobielej masky pre HD rozlíšenie obrazov
mask_rgb.m	skript na vytvorenie farebnej masky pre klasické rozlíšenie obrazov
mask_rgb_hd.m	skript na vytvorenie farebnej masky pre HD rozlíšenie obrazov
nmb_pix.m	skript na vykreslenie závislosti počtu meraní na počte vstupných pixelov
OMP.m	skript s rekonštrukčným algoritmom OMP
rec_all_hd.m	skript na simuláciu jednopixelovej kamery pre obrazy s HD rozlíšením
rec_orig.m	skript na simulácie rekonštrukčnej kvality OMP algoritmu
reconstruction_all.m	skript na simuláciu jednopixelovej kamery pre obrazy s klasickým rozlíšením
transform.m	skript prevedie transformáciu na jednotlivé bloky